

INPUT

Publicación práctica
para usuarios de

Sinclair

Revista mensual 1987

Precio 375 Ptas

Año 2 Número 18

**JUEGA CON
TU MAQUINA
TRAGAPERRAS**

**MAPA Y CARGADOR
DEL LIVINGSTONE**

**TU TAMBIEN
PUEDES HACER
SCROLLS
PROFESIONALES**

**PROGRAMAS
PARA DISEÑO
ASISTIDO POR
ORDENADOR**

...Te seguimos presentando el mejor software del año



Con DANDY vivirás la aventura más complicada que jamás te hayas pensado en una mazmorra. No te será fácil encontrar el tesoro. DANDY es la mazmorra definitiva.

CSA



El universo, objetos tridimensionales, criaturas extrañas y la oscuridad del espacio, llenan de emoción y tensión este juego, donde tu supervivencia depende de tus reflejos.

CSA

CHAMPIONSHIP BASKETBALL

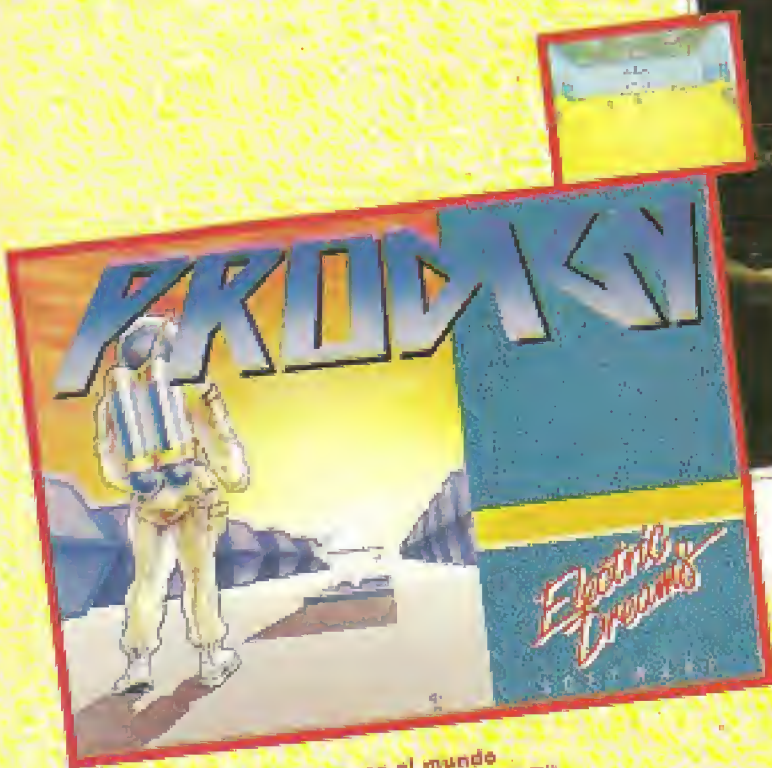
Two-on-Two



GAMESTAR

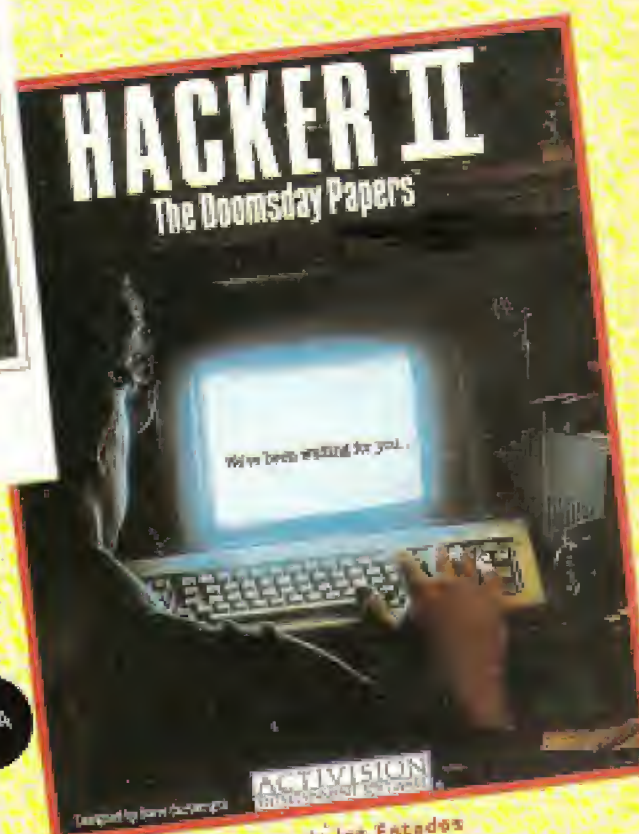
Este BASKET empieza donde otros acaban, porque se basa en el juego de equipo. Con los mejores gráficos de un juego de BASKET y la variedad de posibilidades de juego (Dos jugadores y la computadora, prácticas, liga de 23 jugadores etc.) nunca te cansarás de jugar hasta llegar a ser un campeón.

CSA



PRODIGY nos introduce, en el mundo "MEC" donde debemos conducir a "SOLO" y el hombre sintético que cuida de "NEJO" y librarlo de los peligros más adversos, sin olvidarnos de WARLOCK, el ser mecánico que quiere destruir toda vida orgánica. Sus efectos sonoros y en tres dimensiones lo hacen inmejorable y diferente.

CSA



Saludos del gobierno de los Estados Unidos... La CIA cuenta contigo para proteger a los países de Occidente. Los Rusos tienen en su poder el libro llamado "El día del juicio final". Con él pueden tener el mundo en sus pies. Y aquí estás tú, de lo demás, sólo podemos decirte: BUENA SUERTE. Falta te hace.

CSA

Disponibles para:

COMMODORE
SPECTRUM
AMSTRAD CASS / DISK

CSA

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A: **PROEIN, S.A.**

Distribuido en Cataluña por: DISCOVERY INFORMATICA / Arco Iris, 75 - BARCELONA - Tels. 256 49 08 / 09

Velázquez, 10 - 28001 Madrid - Tels. (91) 276 22 08/09



DIRECTOR: Manuel Pérez

DIRECTOR DE ARTE: Luis F. Bataguer

REALIZACION GRAFICA: Didac Tudela

COLABORADORES: José Vila, Ernesto del Valle, Equipo Molisoli, Ramón Rabaso, Antonio Ibarbel, Jaime Mardones, Carlos Bartolomé

FOTOGRAFIA: Ernesto Waitisch, Joan Boada

INPUT Sinclair es una publicación de
PLANETA-DE AGOSTINI, S.A.

GERENTE DIVISION DE REVISTAS:
Sebastián Martínez

PUBLICIDAD: José Real-Grupo Jota
Madrid: C/ General Varela, 35
Teléf. 270 47 02-03

Barcelona: Avda. de Sarrià, 11-13, 1.^a
Teléf. 250 23 99

FOTOMECANICA: IECFA, S.A.

IMPRESION: Sirven Gráfico
C/ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: B. 38 115-1986

SUSCRIPCIONES: EDISA
López de Hoyos, 141, 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Aníbal, 185, 1.^a
08021 Barcelona

DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Travesera de Gracia, 56, Edificio Odiseo
08006 Barcelona

El precio será el mismo para Canarias que para la
Península y en él está incluida la sobretasa aérea.

**INPUT Sinclair es una publicación
controlada por**



INPUT Sinclair es independiente y no está vinculada a
Sinclair Research o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si
bien la recibe, no responsabilizándose de su pérdida o
extravío. Las respuestas se canalizarán a través de las
secciones adecuadas en estas páginas.

© 1987 by Planeta-De Agostini, S.A.

Copyright ilustraciones del fondo gráfico de Marshall
Cavendish

INPUT

sinclair

SUMARIO

EDITORIAL

4

APLICACIONES

DISEÑO ASISTIDO POR ORDENADOR	14
TRATAMIENTO DE TEXTOS (III)	20
CRUCIGRAMAS EN PANTALLA	24
BETA BASIC 1.8	47

CODIGO MAQUINA

SCROLLS PROFESIONALES	5
------------------------------	----------

PARTICIPA

64 COLUMNAS EN BASIC	39
-----------------------------	-----------

REVISTA DE SOFTWARE

MAPA Y POKES DEL LIVINGSTONE	54
ANALISIS DE NOVEDADES	58

EL ZOCO DE INPUT

65

PROGRAMACION DE JUEGOS (COLECCIONABLE)

LA MAQUINA TRAGAPERRAS FRUTAS Y COCO	31
---	-----------

RESPUESTA POSITIVA

El mes pasado anunciábamos a nuestros lectores la reducción de precios de los videojuegos. Ahora, en el presente número, ya tenemos algunos elementos para hacer balance, bien que provisional y de urgencia, sobre las consecuencias directas que tal medida ha tenido, y se puede pensar que tendrá, sobre esta franja del mercado de software.

No decimos nada nuevo si constatamos que, del lado de la producción y la distribución, el apoyo que recibió la reducción de precios no estaba exento de cierto desasosiego respecto del rumbo definitivo que tomarían los acontecimientos. Tal recelo se justificaba por el temor a que el descenso de precios, y del margen comercial, no fuera contrarrestado por el crecimiento del mercado.

También cabía esperar una cierta reserva psicoló-

gica de parte de los usuarios, habituados a valorar la calidad de los programas en función de su precio.

A nuestro juicio y teniendo siempre en cuenta las limitaciones de perspectiva que mencionábamos antes, los primeros efectos han sido de tono claramente positivo: tanto las empresas como los compradores han asimilado la nueva tarifa.

Sin duda era necesario que los videojuegos se ofrecieran en el mercado a un precio que resultara justo para el comprador, a la vez que asimilable a su poder adquisitivo.

Ésta era una necesidad impostergable si se quería crear un mercado estructurado e integrado en el que nadie tuviera conciencia de ser inadecuadamente tratado. El pasado mes se dio un paso muy importante en ese sentido.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un **ranking** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Aribau, 185. Planta 1. 08021 Barcelona

ELIGE TUS PROGRAMAS

Primer título elegido	Segundo título elegido
Tercer título elegido	Programa que te gustaría conseguir
Qué ordenador tienes	Nombre
1.º Apellido	2.º Apellido
Fecha de nacimiento	Teléfono
Dirección	Localidad
Provincia	

INPUT SINCLAIR N.º 18

RUTINA DE SCROLL

■	RUTINA DE «SCROLL» CON MAPEADO
■	¿QUE ES UN MAPEADO?
■	PRESENTACION
■	EL FORMATO LONCHA

■	CREACION DE LOS GRAFICOS
■	DIRECCIONAMIENTO DE LOS GRAFICOS
■	CODIFICACION DE PANTALLA
■	DEMOSTRACION

Muchas veces habrás querido saber cuál es la manera de realizar presentaciones de tus juegos parecidas a las comerciales, o poder ejecutar rutinas de *scroll* con mapeado. Para eso te será de gran ayuda la rutina que explicamos a continuación y que, además, podrás introducir en tus propios juegos para darles velocidad y dinamismo.

Si te guías por simples referencias este tema te podrá parecer demasiado complejo.

Nada más lejos de la realidad, pues con los artículos que, a partir de este número, vamos a publicar conseguirás buenas rutinas que se ejecutarán con rapidez y, además, al ser reubicables las podrás incorporar a todos tus programas.

Para empezar hemos escogido una rutina de *scroll con mapeado*. Explicándola hablaremos de nociones básicas de código máquina y, desde un punto de vista más práctico, tocaremos los pasos a dar para introducirla en vuestros programas.

Para todos los programas de este artículo hemos utilizado el ensamblador GENS de Hisoft.

Para quienes no dispongáis de él, os recordamos que INPUT publicó en el número anterior su propio ensamblador para Spectrum. Ahora tenéis una buena oportunidad de probar su utilidad y eficacia.

UTILIDAD DE LA RUTINA

La rutina que vamos a comentar realiza un *scroll*, o desplazamiento de la pantalla, *pixel a pixel* hacia abajo. El BASIC del SPECTRUM sólo permite hacer movimientos de la pantalla u objetos, carácter a carácter, o, lo que es lo mismo, de 8 en 8 pixels. El fichero de imagen, *Display File*, tiene 2 modos de pantalla, baja resolución y alta resolución. La baja resolución consta de 0 a 31 caracteres de ancho

por 0 a 23 de alto. Pero el BASIC sólo accede hasta la línea 21, puesto que las líneas 22 y 23 las utiliza para darnos información. Por contra, si usamos la alta resolución disponemos de 0 a 255 pixels horizontales por de 0 a 175 verticales.

Sin embargo, nuestro SPECTRUM sólo tiene una rutina de *scroll*, en baja resolución y solamente hacia arriba. Ésta se halla en la dirección 3582 de la ROM.

La rutina de *scroll* que incorporamos en el mapeado hace un desplazamiento de la pantalla hacia abajo en alta resolución, o *pixel a pixel*.

¿QUE SE ENTIENDE POR UN «MAPEADO»?

Una pantalla del Spectrum utiliza 6912 bytes. Si la usáramos tal y como nos viene sólo podríamos tener en la limitada memoria del ordenador 5 pantallas sin posibilidad de tener un programa. Para resolver este conflicto, los programadores crearon los *mapeados*. El *mapeado* es una rutina dentro de un programa que traba-

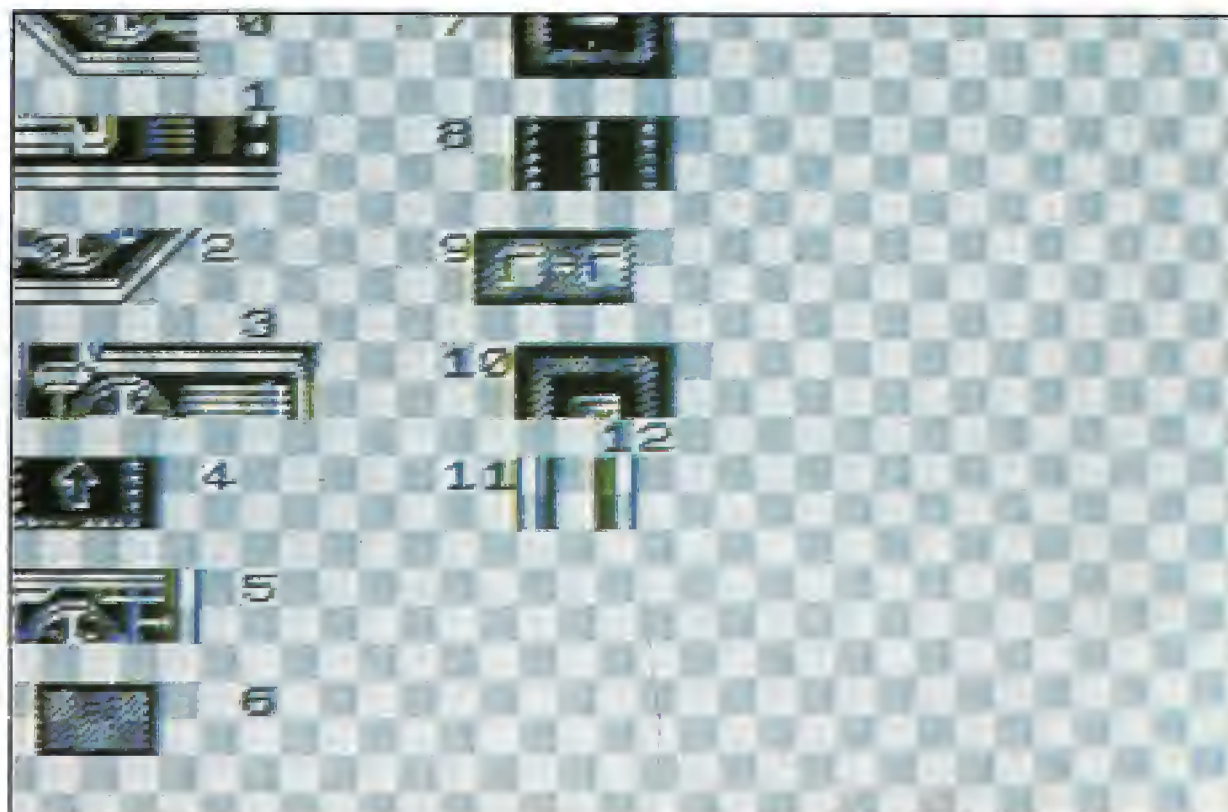
jando con bloques de gráficos, organiza de tal modo la pantalla que nos permite hacer con elementos simples una cantidad enorme de pantallas, ahorrando un volumen ingente de memoria. Una pantalla codificada mediante nuestro mapeado puede ocupar hasta 256 bytes, más incluso que las que contienen pantallas de juegos comerciales.

PROFANATION, juego que todos conocéis, utiliza alrededor de 150 bytes. Imaginaros lo que se puede hacer con 256 bytes...

PRESENTACION

```

10          ORG 50000 ;RUT
              INAS REUBICABLES
20 ;
30 PRESENTACION
40 ;
50          XOR  A
60          OUT (254),A
70          LD   A,70
80          LD   (23624),A
90          LD   HL,22528
100         LD   DE,22529
    
```




```

110 LD BC,768
120 LD (HL),70
130 LDIR
140 LD HL,22528
150 LD DE,22529
160 LD BC,63
170 LD (HL),00
180 LDIR
190 ;

```

La rutina es reubicable, por conveniencia la hemos situado en la dirección 50000. En las líneas 50 y 60 colocamos el BORDER a 0, en las 70 y 80 situamos las dos líneas inferiores con PAPER 0 y INK amarillo. De la 90 a la 130 definimos el PAPER negro y la INK amarilla, de toda la pantalla. De la 140 a la 180 colocamos las dos líneas superiores (0 y 1) en PAPER negro y INK negro. Este camuflaje nos permite imprimir los gráficos en esa zona, sin que aparezcan bruscamente y dando el efecto que el gráfico aparece desde el principio de la pantalla.

Si queréis ver el efecto que realiza el camuflaje, suprimid las líneas desde la 140 a la 180.

RUTINA BUCLEADORA DEL MAPEADO

La subrutina siguiente se encarga de coger la codificación tal y como queremos que aparezca la pantalla. La codificación la analizaremos más adelante.

```

200 MAPSCR
210 ;
220 LD IX,D_PANT
230 LD A,(26999)
240 LD D,00
250 BUC DEC A
260 JR Z,PINTAR
270 LD E,(IX+00)
310 ADD IX,DE
320 JR BUC
330 PINTAR LD A,(IX+00)
335 CP 255
337 RET Z
340 DEC A
350 PINTA PUSH AF
360 LD L,(IX+03)

```

```

370 LD H,00
380 ADD HL,HL
390 LD DE,D_GRAF
400 ADD HL,DE
410 LD E,(HL)
420 INC HL
430 LD D,(HL)
440 LD (23306),DE
450 EX DE,HL
460 DEC HL
470 DEC HL
480 LD E,(HL)
490 INC HL
500 LD D,(HL)
510 LD (23303),DE
520 LD E,(IX+1)
530 LD D,(IX+2)
540 LD (23300),DE
550 LD A,1
560 LD (23302),A
570 PUSH IX
580 PUSH HL
590 PUSH DE
600 CALL IMP_D
610 POP DE
620 POP HL
630 POP IX
640 INC IX
650 INC IX
660 INC IX
670 POP AF
680 DEC A
690 DEC A
700 DEC A
710 JR NZ,PINTA
720 ;

```

En la línea 220 cargamos el registro IX con la dirección de la codificación de la pantalla. En la línea 230 se dimensiona A con el número de pantalla a imprimir o visualizar y se decrementa A (número de pantalla). Si la pantalla a imprimir es la 1 salta inmediatamente a dibujarla. Por contra si no fuese 1 se va al primer valor de la dirección de codificación de pantalla, y se suma a la dirección inicial de pantallas, con lo que se consigue la dirección de la segunda pantalla. Este bucle se realiza hasta encontrar la dirección de la pantalla deseada.

A continuación coge la longitud que tiene la pantalla y la guarda, y realiza

la misma operación con el valor del gráfico que quiere visualizar. Puesto que cada gráfico tiene un número asignado, ese número contiene la dirección de memoria donde se halla ubicado ese gráfico. El siguiente paso es tomar los valores para imprimir en una posición determinada de la pantalla el gráfico en cuestión. Todo esto se realiza entre la línea 350 y la 590. Una vez tiene los valores del gráfico y su posición en la pantalla, salta a la rutina impresora del gráfico que está entre las líneas 1500 y 2120, que más adelante comentaremos. Ésta hace que se imprima el gráfico. A continuación realiza la misma operación hasta que se complete la pantalla. Cuando termina salta a BUCLE.

En nuestro caso, la rutina impresora al estar asociada con el *scroll* sólo opera en las dos líneas superiores.

BUCLE

```

730 BUCLE
740 ;
750 LD B,16
760 BUCA PUSH BC
770 CALL SCROLL
780 POP BC
790 DEC BC
800 DJNZ BUCA
810 LD A,(26999)
820 INC A
830 LD (26999),A
840 CALL MAPSCR
850 RET
860 ;

```

Esta pequeña subrutina es la encargada de montar todo el efecto. En la rutina se aprecia como ésta imprime los gráficos en las dos líneas superiores y acto seguido realiza un BUCLE, el cual desplaza hacia abajo 16 pixels, los impresos con anterioridad en las dos primeras líneas. Una vez realizado el *scroll*, vuelve a imprimir los gráficos siguientes y procede a desplazarlos 16 pixels hacia abajo. El proceso lo ejecuta indefinidamente y no retorna al BASIC hasta que encuentra un comprobante, en nuestro caso 255 indicado en el último gráfico.

RUTINA DE SCROLL EN ALTA RESOLUCION

920 ;RUTINA DE SCROLL EN	1080	PUSH DE	1290 S_TER LD	A,H
ALTA RESOLUCION ABAJO	1090	LD DE,224	1300	SBC A,7
930 ;	1100	SBC HL,DE	1310	LD H,A
940 SCROLL LD HL,22527	1110	EX (SP),HL	1320	JR _5_
950 LD DE,22271	1120	SBC HL,DE	1330 _4_ PUSH HL	
960 LD C,192	1130	EX DE,HL	1340	LD HL,2016
970 _5_ LD B,32	1140	POP HL	1350	ADD HL,DE
980 _2_ LD A,(DE)	1150	DEC C	1360	EX DE,HL
990 LD (HL),A	1160	LD A,C	1370	POP HL
1000 LD A,C	1170	AND 7	1380	LD A,C
1010 CP 2	1180	JR Z,_3_	1390	AND 63
1020 JR NZ,_1_	1190	CP 1	1400	CP 1
1030 XOR A	1200	JR Z,_4_	1410	JR NZ,_5_
1040 LD (DE),A	1210	JR _5_	1420 S_TEROLD	A,D
1050 _1_ DEC DE	1220 _3_ PUSH DE		1430	SBC A,7
1060 DEC HL	1230 LD DE,2016		1440	LD D,A
1070 DJNZ _2_	1240 ADD HL,DE		1450	LD A,C
	1250 POP DE		1460	CP 1
	1260 LD A,C		1470	JR NZ,_5_
	1270 AND 63		1480	RET
	1280 JR NZ,_5_		1490 ;	



Esta rutina produce un desplazamiento de la pantalla hacia abajo una línea cada vez que es llamada. Hay que tener en cuenta, sin embargo, que sólo desplaza la TINTA, no así los ATRIBUTOS. El funcionamiento lo realiza de la siguiente manera: desplaza hacia abajo y una a una las 191 líneas de pixels de la pantalla mediante el BUCLE _5_ y _2_ —no confundirlo con números pues se trata de etiquetas—, que está en su interior. Desplaza cada línea byte a byte borrando la línea continua superior, el decremento de los punteros para cambiar de línea es normalmente 224 o sea, la anchura de la pantalla del Spectrum en alta resolución. Este número surge de restar $256 - 32 = 224$.

Existen algunas excepciones. Cuando la línea es múltiplo de 8, -1 cambia el carácter de origen + 2016. Esto lo realiza en la etiqueta (LABEL) _4_.

Cuando la línea es múltiplo de 8, cambia el carácter de destino -1792. Esto lo realiza en la etiqueta S_tero.

Cuando la línea es múltiplo de 64 cambia el tercio de destino -1792. Esto lo realiza en la etiqueta S_ter.

RUTINA IMPRESORA DE LOS GRAFICOS

Una vez que tenemos almacenados los gráficos en la memoria en formato LONCHA. Atención, no los debemos

confundir con introducción de los gráficos UDG. Este formato consiste en introducir los gráficos en una línea. En UDG cada gráfico está almacenado en memoria mediante una línea de 8 bytes de ancho \times 8 bytes de alto. Para aclarároslo más. Imaginaos que tenemos un gráfico de 3×2 caracteres. La longitud en cada línea en formato LONCHA sería de 24 bytes de ancho \times 16 bytes de alto.

870 ;
880 D_PANT EQU 27000

FIGURA 2

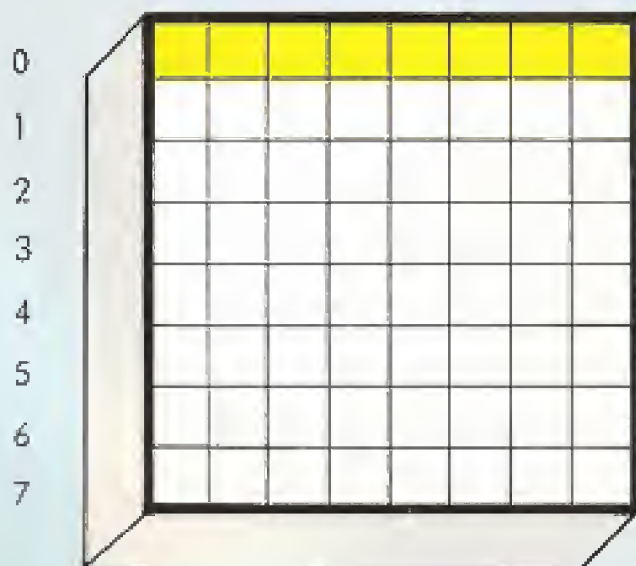
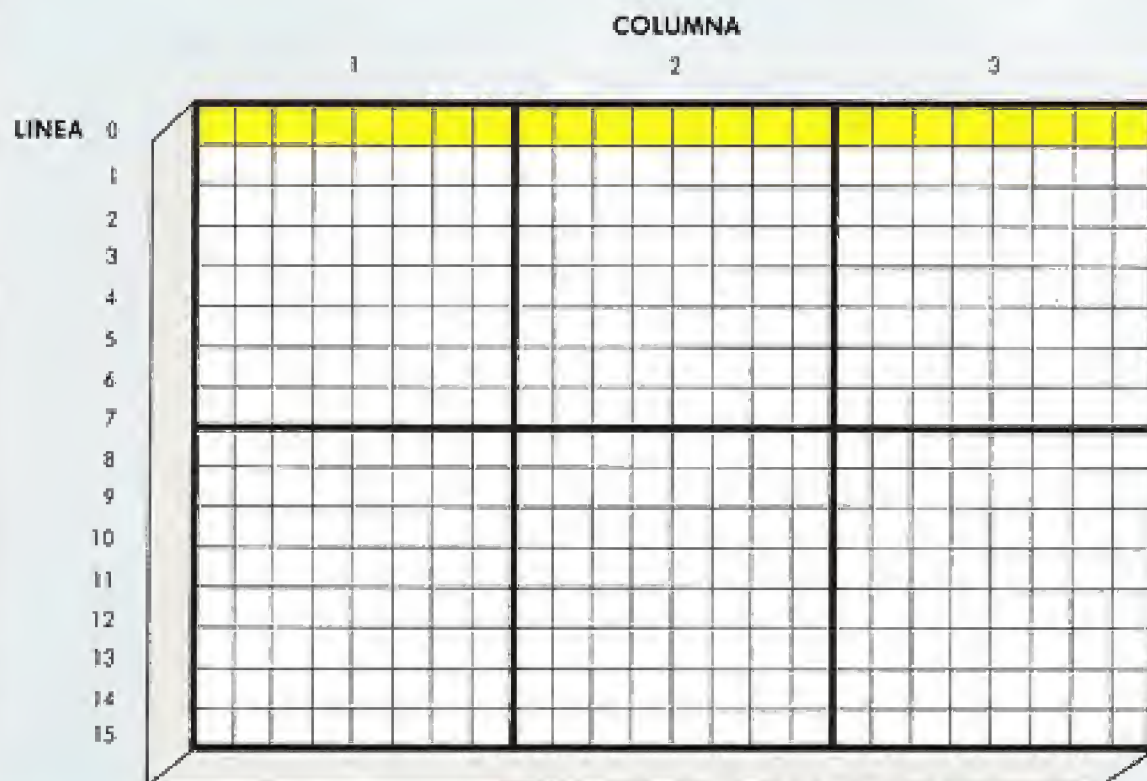


FIGURA 1



En la figura 1 vemos un gráfico de 3 de ancho por 2 de alto. Los datos se almacenan en memoria de la siguiente manera: se coge la línea 0, columna 1, y se «pokea» ese valor. Después el valor de la columna 2, y por último el valor de la columna 3. A continuación se repite la misma operación cambiando de línea hasta que éstas se terminan.

```
890 D_GRAF EQU 32000
900 ;
910 ;
1500 ;RUTINA IMPRESORA DE
      LOS GRAFICOS
1510 ;
1520 IMP_D DI
1530 LD IX,23300
1540 LD BC,(23300)
1550 LD A,B
1560 RRCA
1570 RRCA
1580 RRCA
1590 AND 224
1600 LD L,A
1610 LD A,B
1620 AND 24
1630 OR 64
```

La figura 2 representa un formato normal de 1×1 .


```

1640 LD H,A
1650 LD B,0
1660 ADD HL,BC
1670 LD E,L
1680 LD A,H
1690 RRCA
1700 RRCA
1710 RRCA
1720 AND 3
1730 OR 88
1740 LD D,A
1750 PUSH DE
1760 LD DE,(23306)
1765 LD A,(23304)
1770 _5 EX AF,AF
1780 LD C,8
1790 _2 PUSH HL
1800 LD B,(IX+3)
1810 BIT 0,(IX+2)
1820 JR NZ,_1
1830 LD DE,15616
1840 _1 LD A,(DE)
1850 LD (HL),A
1860 INC DE
1870 INC HL
1880 DJNZ _1
1890 POP HL
1900 INC H
1910 DEC C
1920 JR NZ,_2
1930 LD A,L
1940 AND 224
1950 CP 224
1960 JR Z,_3
1970 LD BC,2016
1980 AND A
1990 SBC HL,BC
2000 JR _4
2010 _3 LD A,H
2020 CP 87
2030 JR NC,_4
2040 LD A,L
2050 AND 31
2060 LD L,A
2070 _4 EX AF,AF
2080 DEC A
2090 JR NZ,_5
2100 POP HL
2110 EI
2120 RET

```

Deberemos acostumbrarnos pues a saber utilizar los gráficos en formato

LONCHA y olvidarnos del formato UDG, si queremos realizar un buen programa ya que éstos nos proporcionan mayor rapidez a la hora de manejar los SPRITES por la pantalla. Todos sabemos que un SPRITE es un conjunto de *pixels* que se mueven por la pantalla y que con una cierta lógica puede darnos la sensación de un movimiento.

Esta rutina es imprescindible para la rutina del mapeado puesto que sin ella no funcionaría nada. La rutina se encarga de leer unos valores escogidos y los coloca en una posición de memoria para posteriormente utilizarlos. La rutina bucleadora de mapeado se encarga de coger los valores de la codificación de la pantalla para que a la hora de llamar a la rutina impresora de los gráficos pueda saber dónde, de cuánto y cómo debe imprimir el gráfico. Más adelante explicaremos cómo usar los parámetros para su correcto funcionamiento.

Concluida la explicación de las rutinas pasamos a explicar detalladamente de qué manera se codifican las pantallas.

CREACION DE LOS GRAFICOS

Lo primero que hay que hacer es crear nuestros propios gráficos, bien sea a mano o con uno de los muchos diseñadores de gráficos, o programas de dibujo. Una vez tenemos los gráficos se les asigna un número a cada gráfico empezando desde 0 hasta concluir con el último gráfico realizado -1. Así si hemos creado 20 gráficos, empezaremos por numerar al primero con 0, al segundo con 1 y así sucesivamente hasta el último. Una vez realizada esta operación comenzaremos a codificarlo en formato LONCHA. Esto lo lograremos con uno de los muchos programas comerciales hechos para tal fin. Una vez codificados todos los gráficos en formato LONCHA, deberemos colocar delante de cada gráfico el ancho y alto de dicho gráfico. Así, si hemos hecho un gráfico de 5 x 2 y éste colocado en 26002, pokearemos: 5 en la posición 26000 y 2 en la posición 26001. Y así con todos los demás gráficos. Una vez salvado este

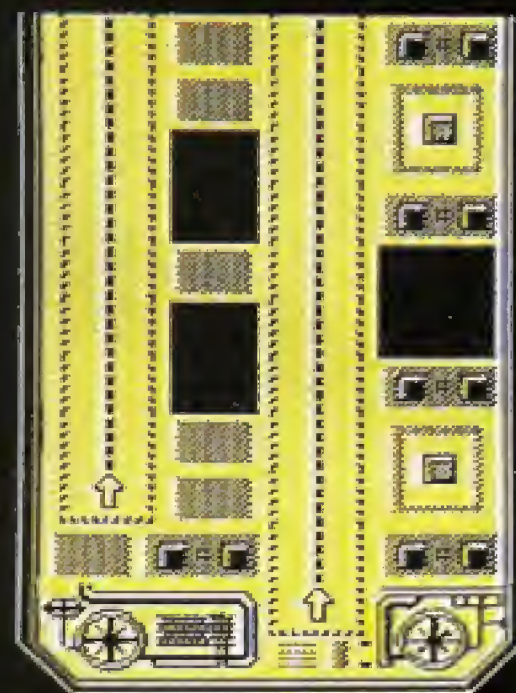
bloque de gráficos, le denominaremos GRAFS:

```

1 REM ***** GRAFS *****
2 LET n=32768: RESTORE
3 FOR i=20 TO 65: LET
  cont=0
4 READ a$
5 FOR j=1 TO LEN a$-1 STEP
  2
6 LET d=16*(CODE
  a$(j)-48-7*(a$(j)>"9"))+
  CODE
  a$(j+1)-48-7*(a$(j+1)>
  "9")
7 POKE n,d: LET cont=cont+d:
  LET n=n+1: NEXT j
8 READ a
9 IF a<>cont THEN GO TO
  14
10 PRINT AT 0,0;"LINEA: ";i
11 NEXT i
12 PRINT AT 0,0;"OK. TODO
  CORRECTO";AT 1,0;"Pulsa
  una tecla para grabar C/M.";
  SAVE "GRAFS"CODE 32768,
  923
13 STOP
14 PRINT "ERROR DE DATA EN
  LA LINEA ";i: PRINT "REVISA
  LAS DATAS": STOP
20 DATA "05028BF680005945
  F6BE1F5922F85D2EBF117
  F5B",2081
21 DATA "36BF08BFAF3
  D00045FD33200022FEC0
  DF70117F3",1852
22 DATA "F3F0008BFC0
  FFF0045FFFFFF0022
  FFFFFFF001100",2794
23 DATA "00000008
  FFFFFFF0040000000000
  200000000001FF",
  1035
24 DATA "FFFF000702555AF2
  FFFFAE30000
  F2FD55F5E3FF",
  3230
25 DATA "FA72FFFFFFA
  FFFFA72FD55F5FF000002
  FFFFAFF",3853
26 DATA "000006FD55F5FFFF
  A7DFFFFFFAE3000203

```


FD55F5",3049	DB6FE492924FFFFFFFFF0	FF",4073
27 DATA "E3FFFFFFFF FFFFFFFFFFFFFFFFFFFF FFFFFFFF",5072	005",3887	55 DATA "FFFE7FE7FCFEFFE 7FEFFFE7FFE7FFFFCFEF FE",4520
28 DATA "FFFF 0000000000000000 FFFFFFFFFFFFFFFF00000000" 2295	44 DATA "02FFFFFFFFFD18000 000051FFFFFFFFFD1C0000 000",2861	56 DATA "7FEFFFE7FFE7FE 7FCFEFFFEFFFE7FFE7 FE",4440
29 DATA "0000000000000000 00000FFFFFFFFFFFFF 000502",1792	45 DATA "51BFCFF6EF51601F F6EF51403FF000514FF03 FEF",2807	57 DATA "7FCFEFFE7FEF0004 02FFFFFFFFFFFFFFFFEAAA AA",3813
30 DATA "4A00016 FD14AF87D6FA24DC4BA1F 444D6CDAFE88",2466	46 DATA "D14FCFCFEFD14FB 037EFD14F4CCBEF914EBC F5E0",3385	58 DATA "AAD5555555E802A 802D3F555F9EA020902D2 0555",2379
31 DATA "40BCF5FD10 404CCBFA206FB037F4400 FCFCFE880",2830	47 DATA "114D7CFAEF914D6 CDAEFD14AF48D6FD14AF8 7D6F",3040	59 DATA "01EA020902D2055 501EA02A902D0055401EA AAAA",1828
32 DATA "FFF03FD100 FFFFFFA200FFFFFF44 000000008800",2663	48 DATA "D1000302FFFFFFFF FFFD55557EAAAABD5555 7EA",3323	60 DATA "AAD5555555FFFFFF FFFFFFFF000402FFFFFF FF",3704
33 DATA "FFFFFF1000000000 20000000004000FFFFFF 8000",1770	49 DATA "AAABD55557EAAA BD55557EAAAABD55557E AAAAB",3221	61 DATA "FFFFFFFFFFFFFFFF 555555FAAAAAAFF55555 5F",3825
34 DATA "0008028BFF2 FFFFFFFFFFFF8BFF2 FFFFFFFFFFFF8B",3581	50 DATA "D55557EAAAABFFF FFFFFFFF000402FBF557 EFF5",3563	62 DATA "FBFFFEFF5FFFDFF FBFFFEFF5FFFDFFBFFFF EF",4956
35 DATA "FF300000000000 78BFF1FFFFFFFF B8B80800000",2401	51 DATA "F4A7DFFBF407EFF5 F007DFFBFFFEFF5FFFDFF FB",4319	63 DATA "F5F007DFFBF7F7EF F5F407DFFBF557EFF5F4A 7DF",4119
36 DATA "00000 D8B7760000000000 68A003FFFFFFFFFE28A", 1958	52 DATA "FFFEFF5FFFDFFB FFFEFF5FFFDFFAAAAAF F5",4715	64 DATA "0001028B8B8B8B8 B8B8B8B8B8B8B8B8B8B8 B8B00",2227
37 DATA "003C0FFFFFFFFF 28B7773F3FFFFF 5F28B80EC0DF8",3459	53 DATA "55555FFFFFFFFF FFFFFFFFFFFFFFFFFFFF 00",4345	65 DATA "0102D1D1D1D1D1D 1D1D1D1D1D1D1D1D1D1D 10000",3347
38 DATA "0000328BF7D332F EAAB5728BF7AF3D780000 328B",2347	54 DATA "0402EFFFFEFFFFE 7FFFE7FE7FCFEFFE7FEFF	
39 DATA "F75B36BD5555728 BF75D2EB80000328BF6B E1F5E",2324		
40 DATA "AAB5728BF680005 8000032000402FFFFFFFF E7FE",2627		
41 DATA "7FCFEFFDBFEFFFB DFFFE7F7EFCFEFF18FEFF FD",4534		
42 DATA "BFFFE7FDBFCFEFFD BFEEFFDBFFFE7FC3FCFE FFF",4451		
43 DATA "FFEEFFEEFFEDB6		



DIRECCIONAMIENTO DE LOS GRAFICOS

A continuación realizaremos la tabla número 2, que será la de dirección de cada uno de los gráficos sin contar los valores de ancho y alto. Si el gráfico 0 está en la 26000, en la tabla de dirección de gráficos deberemos introducir 26002 en la dirección donde vayamos a situar dicha tabla. Así, el inicio de nuestra tabla se halla en la 32000, por lo que deberemos colocar en la 32000, la primera dirección del gráfico 0, en nuestro caso 26002. Esta segunda tabla la llamaremos DIRGRAFS.

```
10 BORDER 0: PAPER 0: INK 7:
  BRIGHT 1: CLS
15 RESTORE 20: FOR n=32000
  TO 32029: READ a: POKE n,
  a: NEXT n
20 DATA 2,128,85,128,200,
  128,27,129,158,129,225,
  129,52,130,103,130,170,
  130,237,130,48,131,115,
  131,134,131,0,0,0,0
```

CODIFICACION DE PANTALLA

La tabla número 3 es la generadora de la pantalla, pues combina de tal modo los gráficos que podrán surgir infinitud de pantallas. Lo primero que hay que hacer es codificar las pantallas, para ello vamos a explicar qué es codificar una pantalla. Codificar una pantalla es, una vez realizadas las anteriores tablas GRAFS y DIRGRAFS, crear una pantalla a partir de gráficos codificados en unas determinadas posiciones y con unos colores. En nuestro caso, al ser un *mapeado con scroll*, sólo imprime en las dos líneas superiores, por lo tanto en la codificación de las pantallas sólo varía la posición horizontal y el número de gráficos que queremos imprimir.

```
10 BORDER 0: PAPER 0: INK 7:
  BRIGHT 1: CLS: POKE
  26999,1
15 RESTORE 20: FOR n=27000
  TO 27180: READ a: POKE n,
  a: NEXT n
```

```
20 DATA 10,8,0,0,13,0,1,20,
  0,2,10,8,0,3,16,0,4,20,0,5
21 DATA 19,8,0,11,9,0,6,12,
  0,9,16,0,8,20,0,9,24,0,12,
  19
22 DATA 8,0,11,9,0,4,13,0,6,
  16,0,8,20,0,7,24,0,12,19,
  8
23 DATA 0,11,9,0,8,13,0,6,
  16,0,8,20,0,10,24,0,12,16,
  8,0
24 DATA 11,9,0,8,16,0,8,20,
  0,9,24,0,12,13,8,0,11,9,0,
  8
25 DATA 16,0,8,24,0,12,16,8,
  0,11,9,0,8,13,0,6,16,0,8,
  24
26 DATA 0,12,16,8,0,11,9,0,
  8,16,0,8,20,0,9,24,0,12,
  16,8
27 DATA 0,11,9,0,8,16,0,8,
  20,0,7,24,0,12,19,8,0,11,
  9,0
28 DATA 8,13,0,6,16,0,8,20,
  0,10,24,0,12,19,8,0,11,9,
  0,8
29 DATA 13,0,6,16,0,8,20,0,
  9,24,0,12,255,0,0,0,0,0,0,
  0
```

En nuestra rutina hemos escogido la dirección 27000 para colocar la tabla 3. Esta dirección la podemos modificar en la línea 880. Una posición menos, la 26999, guarda el número de pantalla que queramos imprimir. Esto quiere decir que si tenemos varias pantallas codificadas y pokeamos en la dirección 26999 el valor 5, y llamamos a la rutina bucleadora de mapeado aparecerá la pantalla número 5. En nuestro caso, al estar tan sólo codificadas las dos líneas superiores no aparecerá la pantalla número 5, sino la parte de nuestro gráfico número 5. Teniendo la dirección 27000 como dirección de la codificación de pantallas, ahí está la longitud que tiene la pantalla +1. En la siguiente posición a ésta, ya empezamos a construir la pantalla de la siguiente manera: el primer valor que tenemos que dar es la posición horizontal donde queremos imprimir el gráfico en baja resolución. El segundo valor que tenemos que dar es

la posición vertical donde queremos imprimir el gráfico en baja resolución. Y el tercer y último valor es el número de gráficos que queremos imprimir. Esta operación la realizaremos tantas veces como gráficos queramos en la pantalla.

Si hemos impreso 4 gráficos, la longitud de la pantalla será 4 (gráficos impresos) \times 3 (valores que tiene cada gráfico) = 12 + 1 = 13 longitud de la pantalla. Este valor se pondrá, como hemos explicado anteriormente, en la tabla 3. Y repetiremos esta fórmula tantas veces como pantallas queramos.

Hechas todas las pantallas colocaremos un 255 en la última posición de la última pantalla, comprobante para retornar. Y con esto se termina la manipulación de las rutinas. A esta última rutina la llamaremos CODPANT.

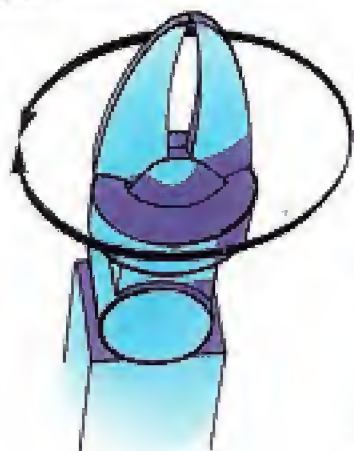
DEMOSTRACION

¿Cómo ejecutar la demostración de la rutina de mapeado? Si disponemos de un ensamblador, teclearemos el listado ensamblador que aparece en estas páginas. Antes de correrlo es mejor salvarlo. Después lo ensamblaremos y bajaremos al BASIC. Ubicado el programa en la dirección 50000 cargaremos los programas GRAFS, DIRGRAFS, y CODPANT independientemente. Cargarlos y ejecutarlos. Una vez en memoria llamaremos a la rutina en código máquina, con un RANDOMIZE USR 50000.

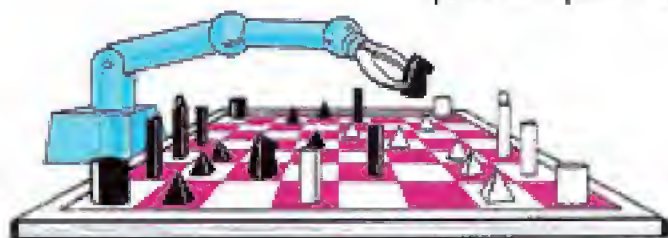
Para volver a repetir el efecto haremos POKE 26999,1: RANDOMIZE USR 50000.

```
1 REM ***** C/M *****
2 LET n=50000: RESTORE
3 FOR i=20 TO 36: LET
  cont=0
4 READ a$
5 FOR j=1 TO LEN a$-1 STEP 2
6 LET d=16*(CODE
  a$(j)-48-7*(a$(j)>"9"))+CODE
  a$(j+1)-48-7*(a$(j+1)>"9")
7 POKE n,d: LET cont=cont+d:
  LET n=n+1: NEXT j
8 READ a
```


!PARTICIPA EN EL CONCURSO!



En INPUT estamos convencidos de que aún puedes hacer muchas más cosas con tu ordenador. Sin duda, muchos lectores estareis utilizando vuestro micro para funciones de lo más variadas, en unos casos; pintorescas, en otros; mientras que algunos listillos habrán podido utilizarlo para resolver tareas complejas. Es lógico, modificando programas y variando los periféricos nuestro ordenador puede prestar sus servicios en infinidad de facetas. INPUT quiere que esas aplicaciones y utilidades a las que has conseguido dedicar tu ordenador, sean conocidas por todos sus lectores y por eso ha organizado el «Concurso de Aplicaciones y Utilidades», en el que puede participar cualquiera de nuestros lectores.



BASES



UTILIDADES Y APLICACIONES: Si tu ordenador controla la calefacción de tu casa, gobierna un robot, dirige un pequeño negocio, organiza la maqueta de tu tren eléctrico, o cualquier cosa interesante u original; envíanos información gráfica y listados de tus programas, grabados en un cassette, diskette o microdrive.

Todo ello habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvideis indicar claramente para qué ordenador está preparado el material, así como vuestro

nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante los próximos tres meses **SORTEAREMOS:**

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
en material microinformático a elegir por los afortunados.

¡No os desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas están revisadas con el máximo interés.

INPUT SINCLAIR
Aribau, 185. Planta 1.ª
08021 BARCELONA

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

DISEÑO ASISTIDO POR ORDENADOR

Dibujos en alta resolución, precisos, limpios, fáciles de realizar con sólo mover un dedo: esto es lo que ofrece el CAD (*computer aided design*; diseño asistido por ordenador). Con el listado que te presentamos aquí, no tienes necesidad de otros programas para obtener gráficos con absoluta facilidad.

Con un software adecuado, las capacidades gráficas de tu ordenador pueden brindarte una nueva y poderosa herramienta de dibujo. No importa si eres un experto en dibujo o más bien un especialista en garabatos, lo importante es que con el listado que aquí te presentamos, tú puedes usar tu ordenador con la misma (e incluso mayor) soltura que el lápiz y el papel.

En la industria, el software CAD (Computer Aided Design) para *mainframes*, grandes ordenadores, permite a sus usuarios dibujar un objeto con todo detalle y simular las respuestas de un determinado diseño a ciertos condicionantes como, por ejemplo, el peso de la estructura, el viento, la vibración o los cambios de temperatura. Es obvio que tales prestaciones superan la capacidad de un ordenador doméstico, pues requieren tanto el procesamiento de una enorme cantidad de datos como una refinada capacidad gráfica para producir distintas vistas del objeto y realizar cálculos.

Pero es frecuente que un delineante o diseñador sólo necesite estudiar o bien un detalle o el aspecto general de la representación visual. En este aspecto, el CAD se convierte en un instrumento que sustituye papel y lápiz en el dibujo técnico, garantizando el trazado perfecto de curvas, rectas y figuras geométricas, con la permanente posibilidad de su corrección o borrado instantáneo.

Este aspecto del CAD sí que puede aplicarse a los microordenadores, tanto si te sientes interesado en los as-

pectos técnicos del dibujo, como si te limitas a realizar pinitos de aficionado. Si bien todos los micros presentan ya sus mandatos de dibujos como parte de su BASIC habitual, elaborar un dibujo detallado requiere muchas líneas de programa.

Lo bueno de este programa de dibujo es que una vez entrado, te permite dibujar lo que desees dentro de las posibilidades gráficas de tu ordenador, sin necesidad de recurrir a ulteriores programas. Te coloca todos los mandatos de dibujo bajo un sencillo y directo control de teclas, de tal modo que lo único que has de hacer es seleccionar la opción o mover el cursor mediante las teclas de flechas. En algunos casos, el programa ofrece mandatos que no se incluyen en el BASIC estándar que incorpora tu máquina.

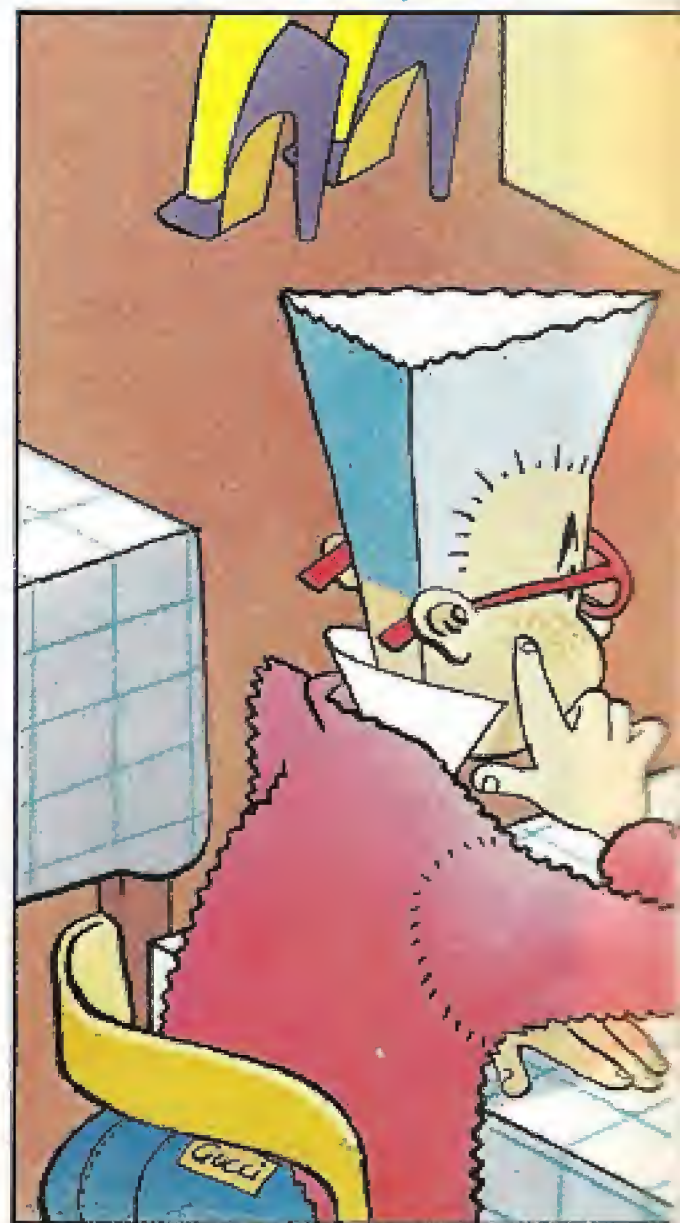
Puesto que las posibilidades gráficas de los distintos ordenadores oscilan bastante de uno a otro, es natural que los programas específicos se diferencien en gran medida, y que cada uno intente sacar el mayor provecho de su potencia, minimizando los puntos débiles.

Así, el Spectrum puede visualizar con facilidad texto y gráficos. Además puede almacenar en varias pantallas distinta información de la que actualmente se está viendo.

Para obtener gráficos de alta resolución desde el BASIC, los usuarios del Commodore 64 necesitan el cartucho BASIC de Simon.

AYUDAS PARA EL DIBUJO

El programa ofrece al usuario un menú o lista de opciones para dibujar ya sean líneas, elipses, círculos o rectángulos. Seleccionando una de ellas y colocando el cursor mediante las teclas de flechas en la posición deseada, puedes crear dibujos realmente espec-



taculares, con la garantía extra de que todas tus líneas serán del todo rectas y tus curvas perfectamente lisas. Y si te equivocas, te concede la merced de borrarlo todo y volver a empezar.

Otras opciones permiten cambiar el color del dibujo, o rellenar áreas con los colores de la paleta que desees. Y una vez concluida la obra de arte, tú puedes optar o bien por guardarla (SAVE) para futuros retoques, o bien hacerla desaparecer y comenzar de nuevo. Esta misma facilidad te permite cargar (LOAD) una figura que no fue dibujada con el presente programa. Por ejemplo, se puede tomar la página del título de un trabajo y modificarla con nuevos dibujos.

- EL ORDENADOR CONVERTIDO EN UN UTENSILIO DE DIBUJO
- COMO AMPLIAR LOS MANDATOS DE GRAFICOS INCORPORADOS
- EMPLEO DEL PROGRAMA

- COMO DIBUJAR ESBOZOS Y DISEÑAR A PULSO
- LA OPCION LINE GARANTIZA EL TRAZADO PERFECTO
- EMPLEO DEL COLOR



trazado de la línea de puntos que desees— pulsa nuevamente SPACE, y podrás mover ahora el cursor sin marcar la pantalla, o bien vuelves al menú para seleccionar otra opción.

La opción LINE funciona como DRAW, sólo que trazando una línea continua y sólida sobre la pantalla; pulsa ENTER para seleccionarla, coloca el cursor en el punto inicial elegido y pulsa SPACE, seguidamente lleva el cursor hasta el punto final y pulsa SPACE de nuevo. La opción seleccionada está disponible siempre que no se cancele seleccionando otra cualquiera. Esto quiere decir que puedes moverte hasta otro punto y dibujar una nueva línea, y otra, etc.

OPCIONES PARA DAR COLOR A LA PANTALLA

Para dar color a la pantalla tienes dos opciones: PAPER te colorea los bordes y el fondo, mientras que INK va dando color a lo que dibujas. Una vez seleccionada una de estas dos opciones, aparecen en la pantalla unas oportunas indicaciones de ayuda para escoger el color. El cursor te muestra el color de la tinta (INK) actual, lo que te permite comprobar instantáneamente el efecto de tu selección. Una vez que has respondido a las indicaciones, la pantalla vuelve al menú, pero el cursor de dibujo permanece donde lo dejaste, por lo que sólo es cuestión de continuar con el dibujo. Se pueden escoger los colores INK tanto antes como mientras se está dibujando. Sólo una vez que hayas salido de la opción de dibujo (pulsando SPACE por segunda vez) la línea se da por definitiva.

Antes de teclear ten en cuenta una cosa importante: este programa contiene lenguaje máquina, por lo tanto asegúrate de guardarlo (SAVE) antes de hacerlo funcionar (RUN):

EMPLEO DEL PROGRAMA

El listado se divide en dos partes. La primera está diseñada de tal forma que funciona con independencia de la segunda, y te permitirá crear dibujos lineales como los que te mostramos. Pero sólo más tarde añadimos algunas de las opciones más sofisticadas del menú.

Al ejecutar el programa (RUN), se te ofrece en pantalla un menú y un cursor. Para seleccionar una opción del menú debes colocar el cursor a la izquierda de la opción, mediante las teclas control Q (arriba), A (abajo), O (izquierda) y P (derecha). Pulsa seguidamente ENTER para efectuar la

selección, el menú desaparecerá y el cursor quedará solo en una pantalla vacía. En cualquier momento se puede o bien volver al menú o abandonarlo pulsando ENTER.

La primera opción del menú es DRAW, para dibujar puntos sobre la pantalla. Para utilizar esta opción, mueve el cursor hasta donde desees iniciar el dibujo, y después pulsa SPACE. A partir de ahora, siempre que muevas el cursor irás dejando un reguero de puntitos a donde quiera que vayas. Si lo desees, puedes acelerar el movimiento manteniendo apretada la tecla SHIFT mientras mueves el cursor. Para salir de la opción DRAW —una vez finalizado el



```

10 BORDER 4: PAPER 7: INK 0:
   OVER 0: CLS
20 POKE 23658,0: LET OP=1
40 DIM O(12): FOR N=1 TO 12:
   READ O(N): NEXT N: LET
   x=127: LET y=87
60 LET x1=x: LET y1=y
65 LET xx=0: LET yy=0
70 FOR n=65368 TO
   65368+73: READ a: POKE n,
   a: NEXT n

```

```

100 RANDOMIZE USR 65380
1020 PRINT INK 9;AT 2,9;"
1030 PRINT INK 9;AT 3,9;"
   1 DRAW
1040 PRINT INK 9;AT 4,9;"
   2 LINE
1050 PRINT INK 9;AT 5,9;"
   3 PAPER
1060 PRINT INK 9;AT 6,9;"
   4 INK
1070 PRINT INK 9;AT 7,9;"

```

```

5 RECTANGLE
1080 PRINT INK 9;AT 8,9;"
6 BOX
1090 PRINT INK 9;AT 9,9;"
7 CIRCLE
1100 PRINT INK 9;AT 10,9;"
8 ERASE
1110 PRINT INK 9;AT 11,9;"
9 OOPS
1115 PRINT INK 9;AT 12,9;"
10 COPY
1120 PRINT INK 9;AT 13,9;"
11 LOAD
1130 PRINT INK 9;AT 14,9;"
12 SAVE
1140 PRINT INK 9;AT 15,9;"
1150 PLOT 72,48: DRAW INK 9;
   111,0: DRAW INK 9;0,
   111: DRAW INK 9;-111,
   0: DRAW INK 9;0,-111
1155 FOR N=1 TO 100: NEXT N
1160 PRINT INVERSE 1: PAPER
   9;AT OP+2,10;">"
1170 PAUSE 0: LET
   a$=INKEY$: IF a$=""
   THEN GO TO 1170
1175 IF a$=CHR$ 13 AND
   OP=9 THEN RANDOMIZE
   USR 65404: RANDOMIZE
   USR 65368: GO TO 1000
1180 IF a$=CHR$ 13 THEN
   RANDOMIZE USR 65392:
   RANDOMIZE USR 65368:
   FOR n=1 TO 100: NEXT n:
   GO TO O(OP)
1190 PRINT AT OP+2,10;" "
1200 IF a$="a" THEN LET
   OP=OP+1: IF OP=13
   THEN LET OP=12
1210 IF a$="q" THEN LET
   OP=OP-1: IF OP=0 THEN
   LET OP=1
1220 GO TO 1160
2000 REM DRAW
2005 FOR n=1 TO 50: NEXT n
2010 GO SUB 8000
2060 IF INKEY$=CHR$ 13
   THEN RANDOMIZE USR
   65380: GO TO 1000
2070 IF INKEY$<>CHR$ 32
   THEN GO TO 2010
2080 FOR n=1 TO 50: NEXT n

```




```

2090 GO SUB 8000: PLOT x,y
2095 IF INKEY$=CHR$ 13
    THEN RANDOMIZE USR
    65380: GO TO 1000
2100 IF INKEY$=CHR$ 32
    THEN GO TO 2005
2110 GO TO 2090
2500 REM LINE
2505 FOR n=1 TO 50: NEXT n
2510 GO SUB 8000
2515 IF INKEY$=CHR$ 13
    THEN RANDOMIZE USR
    65380: GO TO 1000
2520 IF INKEY$<>CHR$ 32
    THEN GO TO 2510
2525 FOR n=1 TO 50: NEXT n
2530 LET xx=0: LET yy=0: LET
    hx=x: LET hy=y PLOT
2540 GO SUB 8000: POKE hx,
    hy: DRAW OVER 1;xx,yy: FOR
    n=1 TO 5: NEXT n: PLOT
    hx,hy: DRAW OVER 1;xx,yy
2550 IF INKEY$<>CHR$ 32
    THEN GO TO 2540
    
```

```

2560 PLOT hx,hy: DRAW xx,yy:
    GO TO 2500
3000 REM PAPER & BORDER
3010 PRINT #1;AT 0,0;"Color
    del Papel (0/7)?"
3015 LET a$=INKEY$
3020 IF a$<"0" OR a$>"7"
    THEN GO TO 3015
3030 POKE 65535,VAL a$*8
3035 RANDOMIZE USR 65416
3040 FOR n=1 TO 50: NEXT n
3050 PRINT #1;AT 0,0;"Color
    del Borde (0/7)?"
3060 LET a$=INKEY$
3070 IF a$<"0" OR a$>"7"
    THEN GO TO 3060
3080 BORDER VAL a$
3090 PRINT #1;AT 0,0;" ";TAB
    31;" ";TAB 31;" "
3095 RANDOMIZE USR 65416:
    RANDOMIZE USR 65380:
    GO TO 1000
3500 REM INK
3510 PRINT #1;AT 0,0:
    
```

```

"Selecciona Tinta (0/7)"
3520 LET a$=INKEY$: IF
    a$<"0" OR a$>"7" THEN
    GO TO 3520
3530 INK VAL a$
3540 PRINT #1;AT 0,0;" ";TAB
    31;" ";GO TO 1000
7500 GOTO 1000
8000 REM RUTINA INKEY$
8005 PLOT OVER 1;x,y
8010 LET a$=INKEY$
8020 IF a$="q" AND y<175
    THEN LET y1=y+1: LET
    yy=yy+1
8030 IF a$="a" AND y>0 THEN
    LET y1=y-1: LET
    yy=yy-1
8040 IF a$="p" AND x<255
    THEN LET x1=x+1: LET
    xx=xx+1
8050 IF a$="o" AND x>0 THEN
    LET x1=x-1: LET
    xx=xx-1
8060 IF a$="Q" AND y<172
    
```



```

THEN LET y1=y+4: LET
yy=yy+4
8070 IF a$="A" AND y>3 THEN
LET y1=y-4: LET
yy=yy-4
8080 IF a$="P" AND x<252
THEN LET x1=x+4: LET
xx=xx+4
8090 IF a$="O" AND x>3 THEN
LET x1=x-4: LET
xx=xx-4
8095 PLOT OVER 1;x,y
8100 LET x=x1: LET y=y1:
RETURN
9000 DATA 2000,2500,3000,
3500,4000,4020,5000,
5500,0,6000,7000,7500
9010 DATA 17,0,64,33,80,
195,1,0,27,237,176,201,
17,80,195,33,0,64,1,0,
27,237,176,201
9020 DATA 17,168,222,33,80,
195,1,0,27,237,176,201,
17,80,195,33,168,222,1,
0,27,237,176,201
9030 DATA 33,0,88,6,4,197,6,
176,203,158,203,166,
203,174,58,255,255,
134,119,35,16,242,193,
16,236,201

```

El anterior programa muestra cómo puedes controlar mediante el teclado la potencia en gráficos de tu ordenador, al tiempo que te permite trazar diseños extremadamente refinados. Pero faltan todavía bastantes funciones si queremos dotar el programa de su potencial total. Falta, sobre todo, la posibilidad de colorear áreas. Habrás visto estas opciones adicionales en el menú, pero hasta ahora no habrás podido acceder a ellas.

Carga (LOAD) el programa anterior, y añádele ahora estas líneas:

```

4000 REM RECTANGULO Y CAJA
4010 LET box=0: GO TO 4030
4020 LET box=1
4030 FOR n=1 TO 50: NEXT n
4040 GO SUB 8000
4050 IF INKEY$=CHR$ 13
THEN RANDOMIZE USR
65380: GO TO 1000

```

```

4060 IF INKEY$<>CHR$ 32
THEN GO TO 4040
4070 FOR n=1 TO 50: NEXT n
4080 LET xx=0: LET yy=0: LET
hx=x: LET hy=y
4090 GO SUB 8000: FOR n=1
TO 2: PLOT hx,hy
4100 DRAW OVER 1;0,yy: DRAW
OVER 1;xx,0: DRAW OVER
1;0,-yy: DRAW OVER 1;
-xx,0: NEXT n
4110 IF INKEY$<>CHR$ 32
THEN GO TO 4090
4120 PLOT hx,hy: DRAW 0,yy:
DRAW xx,0: DRAW 0,-yy:
DRAW -xx,0
4130 IF box=0 THEN GO TO 4030
4135 IF xx=0 THEN GO TO 4040
4140 FOR n=hx TO hx+xx STEP
SGN xx
4150 PLOT n,hy: DRAW 0,yy:
NEXT n
4160 GO TO 4040
5000 REM CIRCLE
5010 FOR n=1 TO 50: NEXT n
5020 GO SUB 8000
5030 IF INKEY$=CHR$ 13
THEN RANDOMIZE USR
65380: GO TO 1000
5040 IF INKEY$<>CHR$ 32
THEN GO TO 5020
5050 FOR n=1 TO 50: NEXT n
5060 LET xx=0: LET yy=0: LET
hx=x: LET hy=y
5070 GO SUB 8000: CIRCLE
OVER 1;hx,hy,ABS xx:
CIRCLE OVER 1;hx,hy,ABS
xx
5080 IF INKEY$<>CHR$ 32
THEN GO TO 5070
5090 CIRCLE hx,hy,ABS xx: GO
TO 5000
5500 REM ERASE
5510 GO SUB 8000
5520 IF POINT (x,y)≠1 THEN
PLOT OVER 1;x,y
5530 IF INKEY$=CHR$ 13
THEN RANDOMIZE USR
65380: GO TO 1000
5540 GO TO 5510
6000 REM COPY
6010 COPY: GO TO 1000

```

```

7000 INPUT "NOMBRE "; LINE
n$: IF LEN n$>10 THEN
GO TO 7000
7010 LOAD n$CODE 50000:
RANDOMIZE USR 65368:
GO TO 1000
7500 INPUT "NOMBRE "; LINE
n$: IF n$="" OR LEN
n$>10 THEN GO TO 7500
7510 SAVE n$ SCREEN$: GO
TO 1000

```

Selecciona la opción RECTANGLE y mueve el cursor hasta una esquina del rectángulo que deseas dibujar, y pulsa SPACE. Muévete hasta la esquina diagonalmente opuesta. Según te mueves va parpadeando el rectángulo sobre la pantalla, de modo que puedes determinar su dimensión con sólo mover el cursor. Pulsa SPACE para fijar su posición.

La opción BOX (caja) funciona como RECTANGLE, sólo que se rellena el área interior.

Otra figura que puedes dibujar es CIRCLE. Mueve el cursor hasta donde desees que esté el centro del futuro círculo, y pulsa SPACE. Desplázate después hasta cualquier punto de la circunferencia descrita y pulsa de nuevo SPACE.

La opción COPY te permite pasar a una impresora ZX la imagen que tienes en pantalla. Selecciona la opción y responde a las indicaciones que aparecen en la pantalla. Una vez realizada la copia, se visualiza el menú.

Para corregir errores o cambiar áreas de la imagen, emplea ERASE o bien OOPS. Puedes borrar pequeños detalles pasando el cursor por encima borrándose un pixel cada vez. Pulsa ENTER para volver al menú.

Si debes hacer cambios más grandes, selecciona OOPS. Se borrará todo lo que hayas hecho desde la última vez que fuiste al menú.

Las dos últimas opciones del menú son SAVE y LOAD (referidos sólo a cassette). Cuando selecciones una de estas dos opciones, te solicitará un nombre para el fichero. Puedes cargar (LOAD) sin especificar el nombre del fichero, pulsando ENTER, pero no así cuando se trata de SAVE.



PROCESADOR DE TEXTOS (III)

Ahora puedes clasificar listados en orden alfabético y buscar una cadena específica, y además imprimir tus propios textos.

En las dos primeras partes del listado del editor de textos habíamos introducido las características básicas del editor de pantalla que te permiten crear archivos de textos o de datos. Esta tercera parte, y final, proporciona las rutinas SORT (Clasificar), SEARCH (Buscar) y PRINTER (Impresora).

CLASIFICAR

La característica SORT emplea una rutina de clasificación de reposición retardada y es utilizada para clasificar líneas de pantalla en orden alfabético. Esta rutina es, por lo tanto, muy útil para realizar listas de clasificación de índices, registros o relaciones documentales.

BUSCAR

La característica SEARCH inspeccionará tu texto para una cadena especificada y puede ser llamada durante el modo de edición. La búsqueda se inicia en el punto donde se encuentra situada la señal > así que hay que asegurarse de que se halle al principio de la copia, para garantizar que se busca todo lo que se encuentre efectivamente en la memoria.

Si una búsqueda no tiene éxito —situación típica en el caso de que una cadena SEARCH haya sido mal teclada— la señal marcadora se colocará al fondo del texto.

Cuando se completa una operación de búsqueda, el programa permanece en el modo de edición, y puedes copiar fácilmente el texto de la cadena SEARCH en el área de trabajo.

IMPRIMIR

La rutina PRINTER (Impresora) te permite producir copias impresas de tus textos archivados. Tiene algunas características especiales, incluyendo una rutina creada para controlar el formateo de la impresora. Si se instala una combinación de impresora que no sea convencional, se deberá, por supuesto, cargar y activar un programa para controlar una interface antes de utilizar la salida a impresión del editor de textos.

FORMATEANDO

En la práctica tiene poco sentido el hecho de cargar y editar tus textos si no puedes imprimirlos en la forma que desees. Por ejemplo, puedes necesitar imprimir la cabecera o el encabezamiento de un documento en el centro de una línea, con un espacio de una línea debajo del texto. Utilizando los comandos de formateo esta tarea se convierte en algo sencillo. Otro ejemplo, muy común, es la escritura de cartas en las que el domicilio del remitente se encuentra cerca del lado derecho de la página y la dirección a la que se dirige la misiva se sitúa al lado izquierdo.

Recuerda que los símbolos que se utilizan han de estar siempre dispuestos al principio de la línea sobre la que actúan.

FUNCIONES DE LOS SIGNOS

El signo # coloca la línea del texto en la parte derecha de la página. Si sólo hay una línea, será colocada de tal forma que su final se encuentre lo más cerca posible de la derecha, como debe ir. Si hay algunas líneas juntas, cada una de ellas con su signo corres-



■	CLASIFICAR Y BUSCAR
■	IMPRIMIR
■	ESCRIBIR
■	FORMATEAR
■	CENTRAR TEXTOS

■	ALINEACION DERECHA
■	ALINEACION IZQUIERDA
■	ESPACIAR TEXTOS
■	PREFIJAR LINEAS
■	SIGNIFICADO DE LOS SIGNOS



pendiente, como las que puedes tener para una dirección o domicilio, el programa medirá la longitud de la línea más extensa y recorrerá todas las otras para igualarlas.

El símbolo & hace que el texto siguiente se inicie en una nueva línea a la izquierda de la página. Este símbolo se utilizará al principio de cada línea de la dirección o domicilio que quieras alinear. El signo del dólar, \$, hace lo mismo, pero deja un espacio de una línea sobre la línea.

El asterisco, *, coloca el texto en el centro de la línea. Cuando se utiliza el asterisco hay que tener cuidado de que el texto no sea demasiado extenso ya que tiene que ser más corto que una línea de texto normal.

```

4000 REM PRINT OUT
4010 LET TT=(PL-LL)/2
4020 LET D=0
4025 FOR N=T+3 TO B-3
4030 LET A$=T$(N)
4032 IF LEN A$=0 THEN NEXT
      N: RETURN
4034 IF A$(LEN A$-1)<>CHR$
      32 THEN GO TO 4037
4035 IF A$(LEN A$)=CHR$ 32
      THEN LET A$=A$( TO LEN
      A$-1): GO TO 4032
4037 LET L=LEN A$
4040 LET C=0
4050 IF C=L THEN NEXT N:
      LPRINT CHR$ 13:
      RETURN
4060 LET C=C+1: LET D=D+1:
      IF C>1 THEN GO TO 4100
4070 IF A$(C)="#" THEN GO
      TO 4500
4080 IF A$(C)="*" THEN GO TO
      4700
4085 IF A$(C)("&" THEN GO TO
      4850
4090 IF A$(C)="$" THEN

```

```

LPRINT CHR$ 13;CHR$
13: LET D=0: GO TO
4900
4100 LET N=N+1: IF N>=B-1
      THEN LET L=LEN A$: GO
      TO 4111
4105 IF T$(N,1)="$" OR T$(N,
      1)="#" OR T$(N,1)="*"
      OR T$(N,1)("&" THEN GO
      TO 4110
4106 LET A$=A$+T$(N)
4107 IF A$(LEN A$-1)<>CHR$
      32 THEN GO TO 4100
4108 IF A$(LEN A$)=CHR$ 32
      THEN LET A$=A$( TO LEN
      A$-1): GO TO 4107
4109 GO TO 4100
4110 LET N=N-1: LET L=LEN
      A$
4111 IF A$(C)=CHR$ 32 THEN
      GO TO 4800
4112 LPRINT A$(C);
4115 IF D>LL THEN LET D=0
4120 GO TO 4050
4500 LET NL=0: LET TA=LL:
      LET BE=0
4510 LET LE=LEN A$-1: IF
      LE>LL THEN PRINT
      FLASH 1;"FORMAT ERROR
      - ADDRESS TOO LONG":
      BEEP 2,10: RETURN
4520 IF LE>BE THEN LET
      BE=LE
4530 LET NL=NL+1: LET
      N=N+1: LET A$=T$(N)
4532 IF LEN A$=0 THEN NEXT
      N: RETURN
4535 IF A$(LEN A$)=CHR$ 32
      THEN LET A$=A$( TO LEN
      A$-1): GO TO 4532
4538 IF A$(1)="#" THEN GO
      TO 4510
4540 LET N=3
4550 LET TR=TT+LL-BE: FOR

```



```

G=1 TO NL: FOR H=1 TO
TR: LPRINT CHR$ 32;;
NEXT H: LET N=N+1: LET
A$=T$(N)
4552 IF LEN A$=0 THEN NEXT
N: RETURN
4555 IF A$(LEN A$)=CHR$ 32
THEN LET A$=A$( TO LEN
A$-1): GO TO 4552
4558 LPRINT A$(2 TO ): NEXT G
4560 NEXT N: RETURN
4700 LET TA=(LL-L)/2+TT: IF
TA<TT THEN LPRINT
CHR$ 13: PRINT FLASH 1;
"FORMAT ERROR -
CANNOT CENTRE": BEEP
2,10: RETURN
4710 LPRINT CHR$ 13;; FOR
M=1 TO TA: LPRINT CHR$
32;; NEXT M: LPRINT
A$(2 TO );: LET D=0:
NEXT N: RETURN
4800 LET SL=LL-D-1: LET
CC=C+1: LET X=1
4810 IF CC>=L THEN GO TO
4825
4820 IF A$(CC)<>CHR$ 32
THEN LET CC=CC+1: LET
X=X+1: GO TO 4810
4825 IF X>=LL THEN LPRINT
CHR$ 13: PRINT FLASH 1;
"FORMAT ERROR - WORD
TOO LONG": BEEP 2,10:
RETURN
4830 IF SL>=X THEN GO TO
4112
4850 LPRINT CHR$ 13;; LET
D=0
4900 FOR M=1 TO TT: LPRINT
CHR$ 32;; NEXT M: GO TO
4050
8000 REM SEARCH

8002 IF Z$="" THEN PRINT
#1;AT 0,0; BRIGHT 1;"NO
TARGET STRING
DEFINED": PAUSE 100:
PRINT #1;AT 0,0;S$:S$:
RETURN
8005 PRINT #1;AT 0,0;S$: IF
P=B-2 THEN LET P=4
8010 FOR N=1 TO 33-LEN
Z$
8020 IF T$(P,N TO N+LEN
Z$)=Z$ THEN LET
N=33-LEN Z$: NEXT N:
GO TO 8050
8030 NEXT N
8040 LET P=P+1: IF P=B-2
THEN LET P=P-1: GO TO
8050
8045 GO TO 8010
8050 LET P=P+1: GO SUB
1000: RETURN

```




```

8500 REM SORT
8505 PRINT #1;AT 0,0;S$;
    S$
8510 LET SS=4
8520 IF T$(SS,1)="7" THEN GO
    TO 8550
8530 LET SS=SS+1: IF SS=B
    THEN PRINT #1;AT 0,0;
    BRIGHT 1;"NO LIMITS
    DEFINED": PAUSE 100:
    PRINT #1;AT 0,0;S$;S$:
    RETURN
8540 GO TO 8520
8550 LET SE=SS+1
8560 IF T$(SE,1)="7" THEN GO
    TO 8600
8570 LET SE=SE+1: IF SE=B
    THEN PRINT #1;AT 0,0;
    BRIGHT 1;"ONLY ONE
    LIMIT DEFINED": PAUSE
    100: PRINT #1;AT 0,0;
    S$;S$: RETURN
8580 GO TO 8560
8600 IF SS=SE-1 OR
    SS=SE-2 THEN GO TO
    8900
8610 PRINT #1;AT 0,0; BRIGHT
    1;"SORTING"
8620 FOR I=SS+1 TO SE-1
8630 LET K=I
8640 FOR J=I TO SE-1
8650 IF T$(J)<T$(K) THEN LET
    K=J
8660 NEXT J: IF I<>K THEN
    LET W$=T$(K): LET
    T$(K)=T$(I): LET
    T$(I)=W$
8670 NEXT I
8900 FOR N=SS TO B: LET
    T$(N)=T$(N+1): NEXT N
8910 FOR N=SE-1 TO B: LET
    T$(N)=T$(N+1): NEXT N:
    LET B=B-2: IF P>B-2
    THEN LET P=P-2
8915 PRINT #1;AT 0,0;S$;
    S$
8930 GO SUB 1000
8940 RETURN

```

Para utilizar la rutina SORT hay que definir primero los puntos de inicio y de final del bloque del texto que se desea clasificar. Para hacer esta

operación teclea un ↑ sobre la primera línea y debajo de la última línea. Para comenzar la clasificación, teclea **[CAPS SHIFT]** y 4. Las dos ↑ desaparecen del archivo de textos durante el proceso de clasificación.

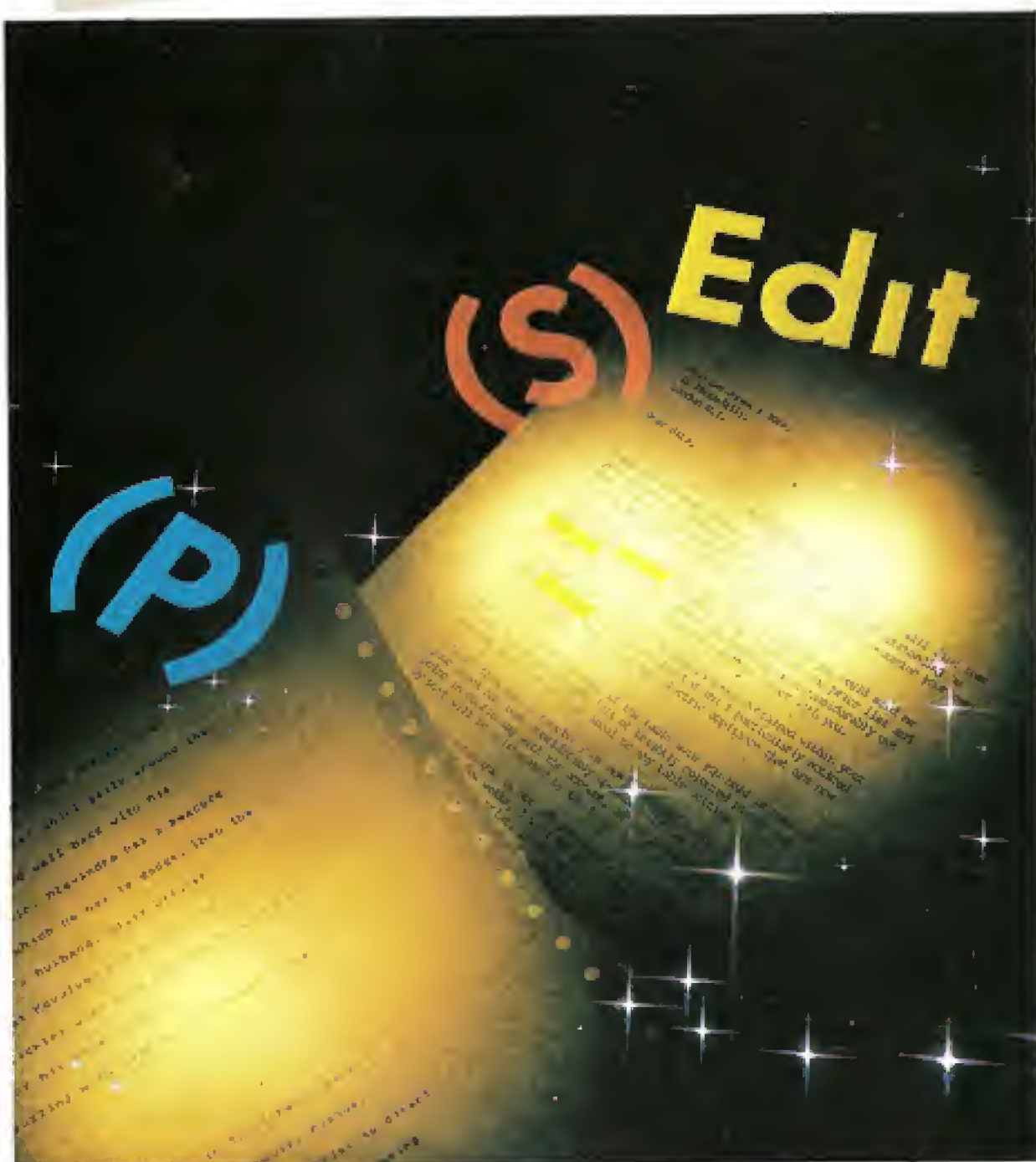
Para poder acceder a la operación SEARCH teclea **[CAPS SHIFT]** y 2, y se te solicitará que entres la cadena SEARCH o cadena de búsqueda. Una vez que hayas introducido la cadena elegida, se iniciará una búsqueda de forma automática. Tan pronto como el ordenador encuentre el elemento a igualar, el cursor aparecerá debajo de su primer caso o ejemplo. Si deseas encontrar más casos o elementos en la cadena, teclea **[CAPS SHIFT]** y 3. Si en cualquier momento deseas redefinir la cadena objetivo, presiona **[CAPS SHIFT]** y 2.

Teclea 6 para enviar el programa a

la rutina de impresión en la línea 4000. Esto hará que el texto se imprima en las posiciones introducidas en la línea 100, esto es, 32 caracteres por línea y 32 líneas por página. Si se desea imprimir en una diferente posición en la impresora hay que teclear 7, en lugar de 6.

Cuando se formatea, se necesita prefijar cualquier línea que se desea imprimir en una forma particular, utilizando un símbolo especial. El signo # prefija todas las líneas que se desean situadas en la parte derecha de la hoja.

El símbolo & fuerza la alimentación de una línea y da inicio a otra en el lado izquierdo del papel. El símbolo del dólar, \$, hace lo mismo, pero en su caso fuerza la alimentación de una doble línea. Utiliza el signo * para centrar el texto.



EL ORDENADOR Y LOS CRUCIGRAMAS

Hoy hemos querido centrarnos en una aplicación del ordenador que, aunque útil, algunos pudieran considerar trivial pero que puede servir de base a otras muchas aplicaciones interesantes.

A todos los aficionados a los crucigramas nos ha ocurrido más de una vez tener que desistir, al cabo de un cierto tiempo, por imposibilidad material de continuar, especialmente cuando el tema es más interesante. ¿Por qué no servirnos del ordenador?

La primera dificultad con la que nos solemos encontrar es el reducido espacio reservado para cada casilla. Luego nos asalta la gran duda: ¿bolígrafo?, ¿lápiz? El lápiz nos permite usar la goma cuando nos equivocamos, pero el bolígrafo se ve mejor, y como vamos a tener que hacer muy pocos cambios, ya que somos unos expertos... La verdad es que siempre caemos en lo mismo, y tras unas cuantas idas y venidas de filas a columnas y de columnas a filas, aderezado con algunos cambios de opinión, lo único que conseguimos es tener aquello confuso e impresentable, a fuerza de meter goma y sobreponer nombres. La desesperación puede llegarnos al comprobar que una palabra que dábamos por segura, y las asociadas que *encajaban perfectamente*, se convierten en un castillo de arena que se nos desmorona al comprobar que la palabra clave es otra y que por lo tanto tenemos que echar abajo gran parte de lo que hemos construido con esfuerzo. La tentación de abandono es muy grande y con frecuencia ocurre que con un sentido práctico tenemos que abandonar. Justo en ese momento hemos cambiado de una situación placentera y estimulante a otra de neta frustración.

Con el programa que presentamos podremos tener un tamaño de casilla cómodo, independiente de las dimen-

siones del crucigrama, ya que hacemos uso de una *ventana* de máximo 15 filas horizontales y 9 verticales. Podremos borrar, modificar, y tantear palabras con gran facilidad, y la presentación siempre será impecable. Incluso podemos archivar lo realizado hasta ese momento y continuar más adelante (es una buena práctica, al menos en los crucigramas de grandes dimensiones, ir haciendo grabaciones periódicas para evitar que un corte de energía o una mala maniobra dé al traste con nuestro trabajo de horas).

Una maniobra que muchas veces se intenta, y casi siempre hay que abandonar por poco práctica, es hacer un crucigrama entre varias personas. Ahora ya no tenemos que renunciar a ese pequeño placer. Las preguntas puede leerlas uno, otro escribirlas y todos ver, opinar y disfrutar de la participación.

Este aspecto pudiera resultar interesante cuando se necesite utilizar el crucigrama como herramienta didáctica en una clase de idiomas, botánica, zoología, etcétera.

Hay quien hace poesía, y también quien escribe crucigramas. Aparte de una gran imaginación se requiere de una base sobre la que escribir y modificar con gran facilidad y seguridad. Nuestro programa no contempla el manejo del texto de las preguntas del crucigrama, ya que alargaría el listado y estamos escasos de espacio. Además, la escritura del texto con ayuda de lápiz y papel (o fichas) no presenta un excesivo problema.

Hasta ahora hemos hablado sólo de crucigramas, pero realmente el programa podemos aplicarlo con igual fortuna a otros juegos a base de paneles de columnas y filas como *salto de caballo*, *damerogramas*, *jerglíficos*, etc. Incluso podemos jugar a las damas.

Considerando que la base de este

programa es el manejo de una gran *pantalla electrónica*, de la cual visualizamos parte a través de una ventana, encontraremos con seguridad docenas de aplicaciones. Una de ellas pudiera ser preparar una *pizarra* con los nombres de una serie de equipos de fútbol, de forma que podamos ir actualizando los resultados de los diversos partidos a medida que vamos teniendo resultados parciales a partir de las informaciones de radio o televisión.

Aun cuando el tema parece inofensivo, el programa reviste cierta complejidad y de él se puede sacar una experiencia apreciable. Como siempre, lo comentaremos con detalle para aquellos que tengan interés en saber el porqué y el cómo. El lenguaje elegido es BASIC, por su difusión, aun cuando evidentemente su manejo sería más rápido en código máquina, haciendo uso de un lenguaje compilado como Pascal o compilando el programa BASIC. En cualquier caso pensamos que en aplicaciones como las que hemos indicado, una gran velocidad de proceso no es vital, ya que unas veces tenemos que esperar que nos llegue el dato que vamos a reflejar y otras tenemos que pensar y darle muchas vueltas en la cabeza antes de decidirnos a escribir algo. De todos modos lo más práctico es poner manos a la obra y ver cómo se comporta.

MANEJO

Una vez tecleado el programa lo ponemos en marcha normalmente (RUN) y la primera pregunta que nos hace es si deseamos trabajar con un crucigrama almacenado en cassette, disco, microdrive o por el contrario queremos comenzar con uno nuevo (¿CARGAR CRUCIGRAMA? [S/N]).

Si la respuesta ha sido «S» el programa nos solicita el nombre del cru-

- UNA BASE PARA NUMEROSAS APLICACIONES
- LOS CRUCIGRAMAS
- UNA GRAN PANTALLA ELECTRONICA

- MANEJO
- MOVIMIENTO DEL CURSOR
- TECLAS ESPECIALES
- COMENTARIOS SOBRE EL PROGRAMA

cigrama (¿NOMBRE DEL CRUCIGRAMA?) y el medio donde le tenemos almacenado (C = cassette, D = disco, M = microdrive). (¿CASSETTE/DISCO/MICRODRIVE? C/D/M.)

Si la respuesta inicial fue «N» (o cualquier letra menos «S»), se nos solicita las dimensiones del crucigrama (¿N.º DE CASILLAS HORIZONTALES?, ¿N.º DE CASILLAS VERTICALES?).

De forma automática el ordenador nos presenta en pantalla el reticulado correspondiente, así como información sobre la situación del cursor. Éste aparecerá inicialmente en la casilla del rincón superior izquierdo (Fila 1, Columna 1). El máximo reticulado es de 15 columnas y 9 filas (se ha elegido esta dimensión para que resulte cómodo el manejo en pantalla). Si las dimensiones del crucigrama son mayores, la parte inicial a visualizar corresponderá a las 15 columnas y 9 primeras filas y aparecerá una información complementaria indicando las dimensiones máximas del crucigrama (ej. 1/56 F=22 C=35 1/75, indicará que estamos moviéndonos en un crucigrama de 1-56 filas y 1-75 columnas y que estamos operando en este momento en la casilla de cruce entre la fila 22 y la columna 35). Si las dimensiones de nuestro crucigrama son inferiores, en una o las dos direcciones, el cuadrulado quedará centrado en la pantalla de forma automática, y sobre él sólo aparecerá información de la fila y columna activa.

Una vez dibujada la retícula del crucigrama se produce una pequeña pausa hasta que aparece el cursor parpadeante. Esta pausa depende del tamaño del crucigrama en la pantalla. Aunque no se aprecia a simple vista, realmente se está explorando el contenido de todos los caracteres del fichero (matriz) que corresponden a la



pantalla que estamos visualizando (hasta que hayamos introducido caracteres se imprime «nada» sobre «blanco») y este bucle en BASIC lleva su tiempo, como indicábamos al comienzo. Tan pronto aparece en pantalla el cursor ya tenemos control sobre el programa.

MOVIMIENTO DEL CURSOR

Hay que distinguir entre moverse «dentro de la pantalla» y «dentro del crucigrama». La pantalla es una ventana a través de la cual nos movemos «sobre» el crucigrama, pero el efecto es el de tener una ventana «fija» detrás de la cual se desplaza el crucigrama (lógicamente esto sólo tiene sentido cuando las dimensiones del crucigrama son mayores que la ventana en pantalla). El resultado es que cuando nos queremos salir de la pantalla por la derecha, aparentemente no conseguimos nuestro propósito y por el contrario la información de la pantalla se «desliza» hacia la izquierda. Cuando nos salimos fuera del marco de la ventana, el programa ne-

cesita situar en pantalla cada una de las nuevas posiciones, procedentes del fichero (matriz), y por tanto sólo reaparecerá el cursor pasado un tiempo, tal como ocurría la primera vez. El programa interpreta que se desea cambiar de ventana cuando queremos forzar una salida en cualquiera de las direcciones, aun cuando esto no sea posible por el dimensionado (se vuelve a leer la misma información como si fuese distinta). Se maneja así por facilidad de programa, pero con unas pocas instrucciones más el lector podrá adaptar el programa para que se distinga entre ventana igual tamaño que el crucigrama y ventana menor que el crucigrama. La diferencia son unos pocos segundos y hemos preferido dejar las cosas más sencillas. Por otro lado, simplemente con tener un cierto cuidado en el manejo de las casillas marginales tal como se indica más adelante, también se obvia esta pequeña demora.

Las teclas para mover el cursor en las cuatro direcciones son las correspondientes a las flechas (Caps Shift + flechas). Tan pronto rebasemos (o lo

intentemos) el límite de la cuadrícula se producirá el cambio a otra ventana, como ya se ha indicado. A fin de no perder el contexto general, cada nueva ventana sólo se desplaza hasta un máximo de 8 columnas y 5 filas nuevas. En los cambios de pantalla el cursor reaparece en el extremo de la fila o columna por donde salió.

Existe otro juego de teclas (Symbol Shift + flechas) para desplazarse directamente entre ventanas. Esto resulta útil cuando se desea manejar informaciones bastante alejadas dentro del entorno del crucigrama.

Cuando aún no hemos realizado ninguna inscripción, el cursor está en el rincón superior izquierdo. Si en este momento pulsamos «A», esta letra aparecerá bajo el cursor parpadeante. Si pulsamos «C», esta letra aparecerá en el lugar de la primera. Si ahora pulsamos «Caps Shift + →» el cursor se desplazará a la derecha un lugar. Al pulsar nuevamente «A», ya no aparecerá superpuesta a «C», sino en el lugar del cursor, pero éste se desplazará automáticamente un lugar hacia la derecha, en espera de una nueva letra, y así sucesivamente hasta que se altere la dirección mediante otra flecha (hacia abajo o hacia la izquierda, en cuyo caso la escritura será automática en ese sentido) o hasta que se trate de pasar el límite de la carátula, en cuyo caso se producirá un cambio automático de ventana. Si deseamos que esto no ocurra, cuando el cursor esté, por ejemplo, en la última casilla de la derecha, pulsaremos «Control Shift + «#»» y estaremos en una situación análoga a la inicial (las letras irán apareciendo sucesivamente, sin producirse avance, hasta que se especifique otra dirección).

TECLAS ESPECIALES

Se han programado tres letras con las funciones que se detallan (el lector no encontrará dificultad en programar otras letras o caracteres modificando la rutina correspondiente, explicada más adelante):

- * - Control Shift + «_» para «Ñ»
- * - Control Shift + «@» para «Ü»
- * - Control Shift + «£» para «■»

Una vez que situemos los cuadros negros en las casillas correspondientes, ya estamos en condiciones de ponernos a jugar.

GRABACIÓN EN CASSETTE/DISCO

En cualquier momento podemos realizar la grabación en soporte magnético del contenido del crucigrama (Symbol Shift + «↑») y continuar. Esta opción está pensada tanto para el caso de archivar un crucigrama terminado como para el archivo de las diversas etapas, por seguridad o para continuar en otro momento.

COMENTARIO AL PROGRAMA BASIC

Comentaremos aquellos bloques funcionales de mayor interés.

Se genera una matriz de las dimensiones requeridas (DIM A\$(DVMC, DHMC,1), línea 480) cuyos parámetros se introducen mediante INPUT en las líneas 350/360. En dicha matriz se irán alojando los caracteres de las diversas casillas, de manera que exista una relación biunívoca entre casillas del crucigrama y celdas de la matriz.

Para aprovechar al máximo la pantalla del Spectrum se ha dimensionado una máscara de crucigrama que ocupe 31 columnas y 19 filas, teniendo en cuenta que se requieren dos renglones para alojar la información operativa sobre dimensionado total, posición del cursor en la pantalla y posición de la casilla activa en el crucigrama (líneas 810-910). No obstante ambos valores pueden reducirse si se precisa, ya que están parametrizados (ver línea 410). Cuando las dimensiones del crucigrama no requieren toda la superficie de la pantalla, la máscara se reduce a la dimensión precisa y a la vez se centra en la pantalla de manera automática con ayuda de las rutinas de las líneas 390-460 y 560-640. Cada carácter ocupa el centro de una casilla y se halla rodeado por espacios vacíos correspondientes a medio carácter.

La lectura de todos los datos correspondientes a una pantalla se realiza mediante la rutina de la línea 740.

El control del movimiento del cur-

sor se realiza mediante las rutinas de las líneas 930-1500. Como puede observarse el procedimiento es un tanto «farragoso». En líneas generales se trata de coordinar la posición del cursor en la pantalla, analizando en cada caso la tecla pulsada (detección mediante línea 970) para analizar si se trata de tecla de carácter o tecla de desplazamiento, con la posición correspondiente en la matriz donde se aloja el contenido de cada casilla del crucigrama. La rutina analiza igualmente si se traspasan los límites de la máscara en pantalla, para decidir cuál es la siguiente parte de crucigrama a explorar.

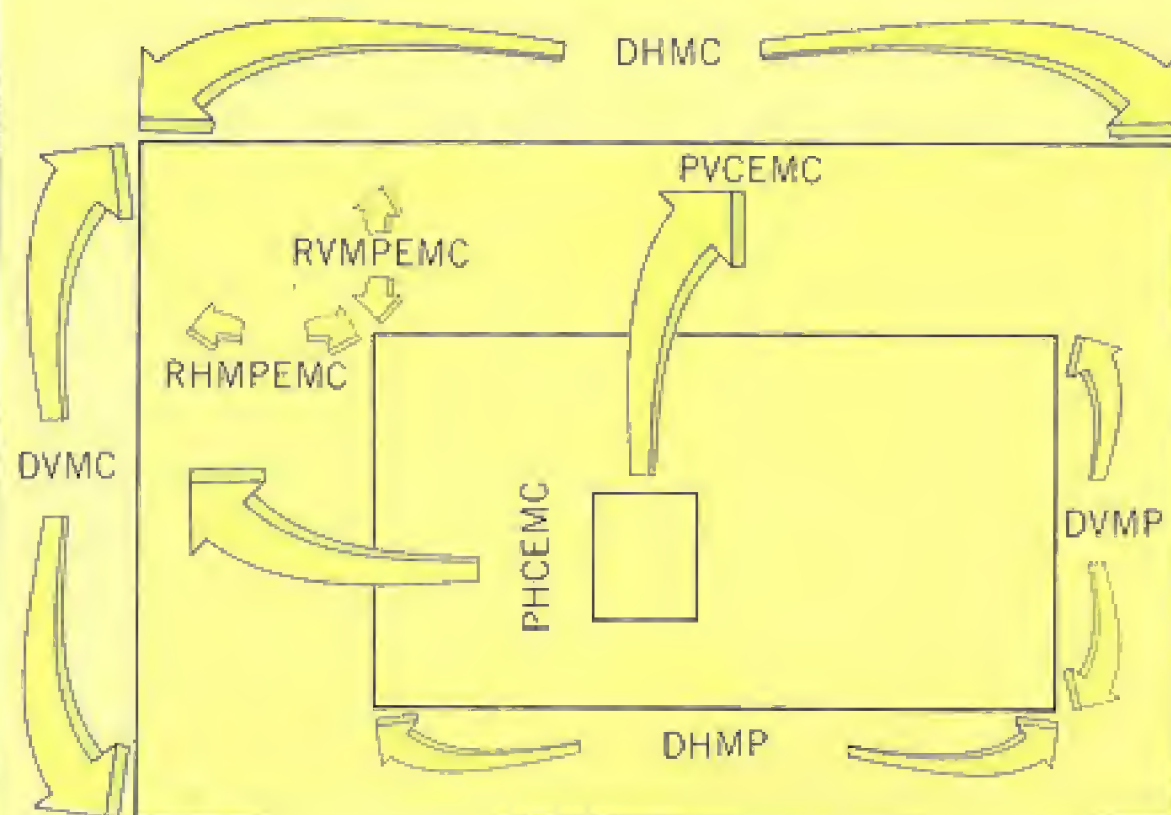
Si el carácter pulsado es «Symbol Shift + «↑»» se detecta en línea 1022 y se ejecuta la rutina de grabación (líneas 2000-2035). Mediante esta rutina se archivan dos matrices :N() y A\$(). En la primera de las cuales se almacenan las variables que definen las dimensiones del crucigrama y en la segunda figuran los caracteres de cada casilla del mismo. El nombre de este fichero es el elegido por nosotros (NS en línea 2010). Para el primero se utiliza el mismo nombre pero añadiéndole automáticamente «.D». Ésta es la razón de que sólo se permitan 8 caracteres en el título en lugar de los 10 que indica el manual (ver línea 2011).

En el caso de querer partir de un crucigrama ya almacenado en cassette, disco o microdrive, el proceso es exactamente el mismo pero a la inversa (líneas 318/320, 490/495). Obsérvese que la carga de A\$() se realiza lógicamente después de su dimensionado (línea 480), realizado con los datos cargados anteriormente en N(). El flag «K» (0/1) define los dos modos de actuación (líneas 310 y 370).

En este programa resulta de capital importancia el manejo de todas aquellas variables que afectan, sobre todo, a la situación de los datos en la pantalla y en la matriz del crucigrama, y por ello vamos a referenciarlas. En las dos figuras puede verse su significado de una manera más gráfica.

Por la estructura del programa se pierde la posibilidad de cambio entre los diversos modos del ZX-Spectrum, ya que sólo se permiten entradas

FIGURA 1



Ventana en Matriz de Crucigrama

DHMC=... Dimensión Horizontal de la Matriz del Crucigrama
 DVMC=... Dimensión Vertical de la Matriz del Crucigrama
 DHMP=..... Dimensión Horizontal de la Matriz de Pantalla
 DVMP=..... Dimensión Vertical de la Matriz de Pantalla
 RHMPEMC= Ref. Horiz. de Matriz Pantalla En Mat. Crucigrama
 RVMPEMC= Ref. Vert. de Matriz Pantalla En Mat. Crucigrama

(INKEY\$) de un solo carácter y para el paso a «L», «C», «G» se requiere una entrada combinada. Por esta razón habría que tener la precaución de arrancar siempre en modo «C» y además no se tiene acceso al uso de «UDG's», necesarios para las letras o caracteres que no existen en el teclado inglés como pueden ser «Ñ» o «Ü». Igual problema tendríamos para hacer uso de los caracteres gráficos (ej. cuadrado negro).

Con la línea 218 (POKE 23658,8) conseguiremos situarnos en modo «C» (mayúsculas), independientemente del modo en que estuviésemos al arranque.

En las líneas 200-215 tenemos alojado el procedimiento para reprogramar ciertas teclas de caracteres estándar, normalmente imposible por en-

contrarse en ROM. Detallaremos los diversos pasos:

- 1) Copiamos el juego estándar de caracteres en ROM a partir de la posición de memoria 6000 con ayuda de la siguiente rutina:

Ensamblador	C/M
LD HL,15360	33,0,60
LD DE,60000	17,96,234
LD BC,2048	1,0,8
LDIR	237,176
RET	201

Esta rutina se carga a partir de la posición 65000 mediante el bucle de DATA de las líneas 202/205 y se rueda con RANDOMIZE USR 65000 (línea 207).

- 2) Con el bucle de la línea 211 se leen las DATA de la línea 213, correspondientes al nuevo con-

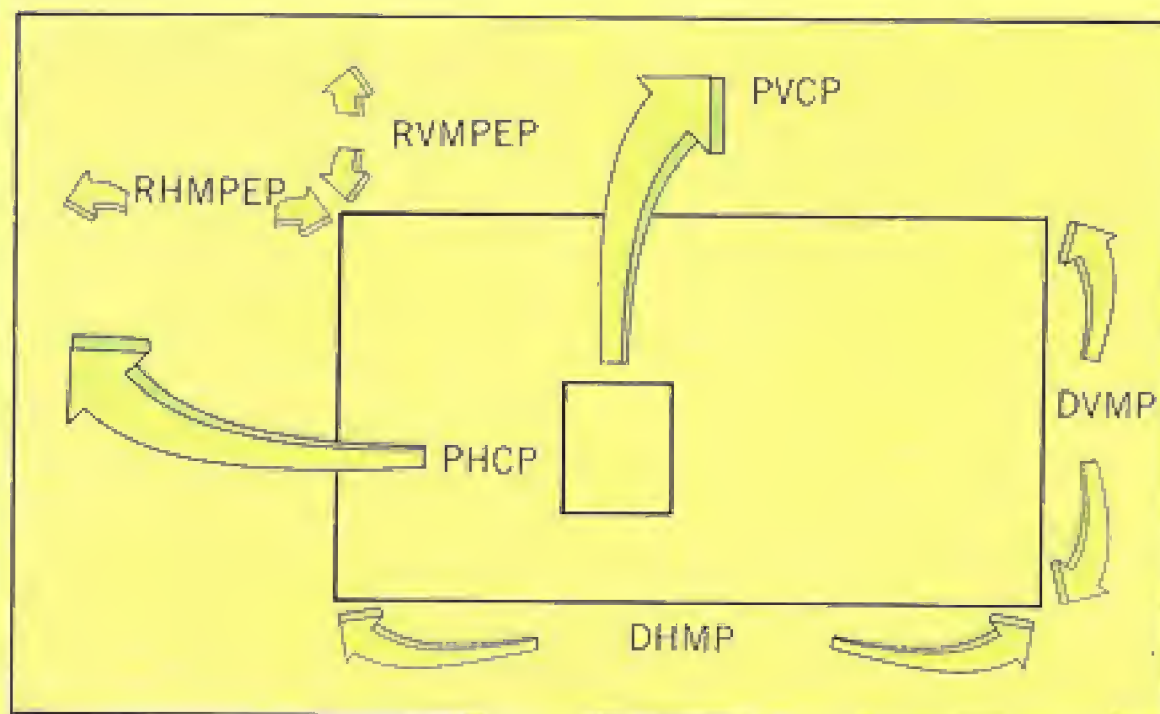
tenido de los caracteres de código ASCII 64, 95, 96. «@» pasa a ser «Ñ», «_» pasa a «Ü» y finalmente «f» a «■».

- 3) Mediante los POKE de la línea 215 instruimos al ordenador para que a partir de este momento utilice el nuevo juego de caracteres.

```

10 REM PROGRAMA PARA
    HACER CRUCIGRAMAS
15 REM
20 REM *****
30 REM *   MOLISOFT 87   *
40 REM *****
45 REM
50 REM DEFINICION DE
    VARIABLES
60 REM
65 REM DHMC=
    DIM.HORIZ.MAT.CR
67 REM DVMC=
    DIM.VERT.MAT.CR
69 REM DHMP=
    DIM.HORZ.MAT.PA
71 REM DVMP=
    DIM.VERT.MAT.CR
73 REM RHMPEMC=REF.H.MA
    T.P.E.C.
75 REM RVMPEMC=REF.V.MAT
    .P.E.C.
77 REM RHMPEP=
    REF.H.MAT.P.E.P.
79 REM RVMPEP=
    REF.V.MAT.P.E.P.
81 REM PHCP= POS.H.CUR.EN
    PA
83 REM PVCP= POS.V.CUR.EN
    PA
85 REM PHCEMC=
    POS.H.CUR.EN CR
87 REM PVCEMC=
    POS.V.CUR.EN CR
190 REM
200 REM CARACTERES
    ESPECIALES
201 REM
202 CLEAR 59999: FOR N=0 TO
    11: READ A: POKE
    65000+N,A: NEXT N
205 DATA 33,0,60,17,96,234,
    1,0,8,237,176,201
    
```


FIGURA 2



Ventana en Pantalla

RHMPEP= Ref. Horiz. de la Matriz de Pantalla en Pantalla
 RVMPEP= Ref. Vert. de la Matriz de Pantalla en Pantalla
 PHCP=..... Posición Horizontal del Cursor en Pantalla
 PVCP=..... Posición Vertical del Cursor en Pantalla
 PHCEMP=.. Posición Horiz. del Cursor en Matriz de Pantalla
 PVCEMP=.. Posición Vert. del Cursor en Matriz de Pantalla

```

207 RANDOMIZE USR 65000
211 FOR M=1 TO 3: READ A:
    FOR J=0 TO 7: READ W:
        POKE (60000+A*8+J),W:
    NEXT J: NEXT M
213 DATA 64,66,0,66,66,66,
    66,60,0,95,60,0,98,82,
    74,70,66,0,96,255,255,
    255,255,255,255,255,255
215 POKE 23606,96: POKE
    23607,234: REM CAMBIO
    AL NUEVO JUEGO DE
    CARACTERES
216 DIM N(2): LET X$="R"
217 BORDER 6: PAPER 6:
    BRIGHT 1: INK 0: CLS
218 POKE 23658,8: REM
    CAMBIO A MAYUSCULAS
219 REM
    
```

```

220 REM CARGA DE
    CRUCIGRAMA DESDE
    CASSETTE/DISCO/
    MICRODRIVE
225 REM
240 CLS : PRINT #0: PAPER 7:
    BRIGHT 1: INK 1:"CARGAR
    CRUCIGRAMA ? (S/N)"
250 LET P$=INKEY$: IF P$=""
    THEN GO TO 250
260 IF P$<>"S" THEN GO TO
    330
310 CLS : LET K=1: INPUT
    PAPER 7: BRIGHT 1: INK 1:
    "NOMBRE DEL
    CRUCIGRAMA":N$: IF
    N$="" THEN GO TO 310
312 IF LEN N$>8 THEN PRINT
    #0: PAPER 1: BRIGHT 1:
    
```

```

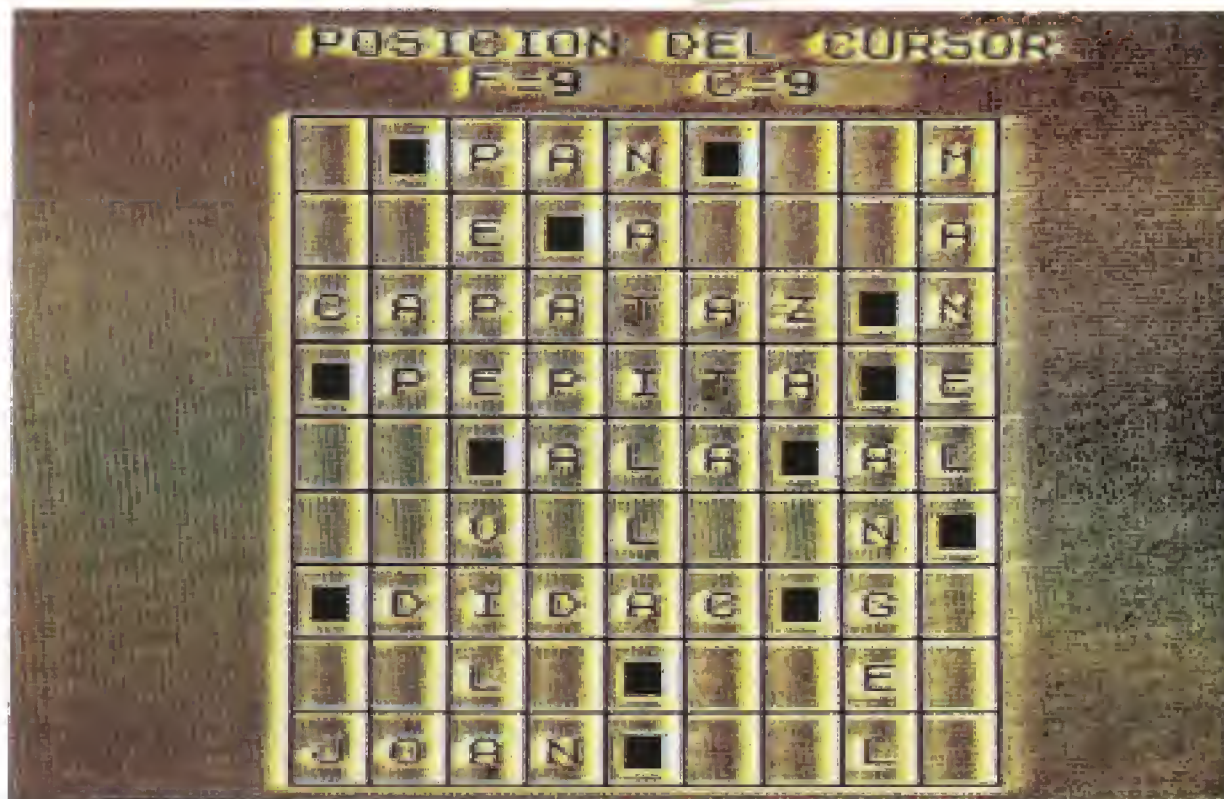
    FLASH 1: INK 7:"NOMBRE
    MUY LARGO (Max. 8)":
    PAUSE 0: GO TO 310
315 PRINT #0: PAPER 7:
    BRIGHT 1: INK 1:
    "CASSETTE/DISCO/
    MICRODRIVE? C/D/M":
318 LET X$=INKEY$: IF
    X$="C" OR X$="D" OR
    X$="M" THEN GO TO 320
319 GO TO 318
320 CLS : IF X$="D" OR
    X$="M" THEN LOAD "*"m":
    1:N$+".D" DATA N()
322 IF X$="C" THEN LOAD
    N$+".D" DATA N()
325 LET DHMC=N(1): LET
    DVMC=N(2): GO TO 390
327 REM
330 REM DIMENSIONADO
    INICIAL DEL CRUCIGRAMA
340 REM
350 INPUT PAPER 7: BRIGHT 1:
    INK 1:"No. CASILLAS
    HORIZONTALES ";DHMC
360 INPUT PAPER 7: BRIGHT 1:
    INK 1:"No. CASILLAS
    VERTICALES ";DVMC
370 LET K=0
380 REM
390 REM DIMENSIONADO
    MATRIZ CRUCIGRAMA
400 REM
410 LET L=31: LET A=19: REM
    CASILLAS UTILES EN LA
    PANTALLA
420 IF DHMC>(L-1)/2 THEN
    LET DHMP=L
425 IF DHMC<=(L-1)/2 THEN
    LET DHMP=DHMC*2+1
430 IF DVMC>(A-1)/2 THEN
    LET DVMP=A
435 IF DVMC<=(A-1)/2 THEN
    LET DVMP=DVMC*2+1
440 LET RHMPEMC=0: LET
    RVMPEMC=0
450 LET RHMPEP=INT
    ((L-DHMP)/2): LET
    RVMPEP=2+INT
    ((A-DVMP)/2)
460 LET PHCP=RHMPEP+2:
    LET PVCP=RVMPEP+2
    
```



```

480 DIM A$(DVMC,DHMC,1)
490 IF K<>0 AND (X$="M" OR X$="D") THEN LOAD "*"M";
      1;N$ DATA A$()
495 IF K<>0 AND X$="C" THEN LOAD N$ DATA A$()
500 GO SUB 560
520 GO SUB 720: GO SUB 930: GO SUB 1190
540 GO TO 510
550 REM
560 REM MASCARA CRUCIGRAMA
570 REM
580 CLS : PRINT AT (RVMPEP-1),8;"POSICION DEL CURSOR"
590 IF DHMC>(L-1)/2 OR DVMC>(A-1)/2 THEN PRINT AT RVMPEP,2;"1/";
      DVMC;TAB 12;"F=";TAB 18;"C=";TAB 27;"1/";DHMC
600 IF DHMC<=(L-1)/2 OR DVMC<=(A-1)/2 THEN PRINT AT RVMPEP,12;"F=C="
610 LET PX=((RHMPEP+1)*8+4): LET PY=192-((RVMPEP+4)*8-4)
620 FOR I=0 TO DVMP-1 STEP 2: PLOT PX,PY-I*8: DRAW ((DHMP-1)*8),0: NEXT I
630 FOR I=0 TO DHMP-1 STEP 2: PLOT PX+I*8,PY: DRAW 0,-((DVMP-1)*8): NEXT I
640 REM
720 REM PRESENTACION EN PANTALLA (15 x 9 CUADROS CRUCIGRAMA ORIGINAL
730 REM
740 FOR V=1 TO (DVMP-1)/2: FOR H=1 TO (DHMP-1)/2: PRINT AT (RVMPEP+V*2),(RHMPEP+H*2);A$(RVMPEMC+V,RHMPPEMC+H): NEXT H:NEXT V
750 RETURN
760 REM
810 REM INFORMACION CURSOR
820 REM
830 PRINT AT RVMPEP,14; BRIGHT 0;" ";AT RVMPEP,14; BRIGHT 0;PVCEMC
840 PRINT AT RVMPEP,20; BRIGHT 0;" ";AT RVMPEP,20; BRIGHT 0;PHCEMC
850 RETURN
860 REM
870 REM CALCULO CASILLA ACTIVA
880 REM
890 LET PHCEMC=(PHCP-RHMPEP)/2+RHMPPEMC
900 LET PVCEMC=(PVCP-RVMPEP)/2+RVMPEMC
910 RETURN
920 REM
930 REM MOVIMIENTO DEL CURSOR
940 REM
945 LET IH=0: LET IV=0
950 LET V=PVCP: LET H=PHCP: GO SUB 870: GO SUB 810
960 PRINT AT PVCP,PHCP: FLASH 1:A$(PVCEMC,PHCEMC)
970 LET C$=INKEY$: IF C$="" THEN GO TO 970
990 IF CODE C$=11 THEN LET PVCP=(PVCP-2): LET IH=0: LET IV=-2: IF PVCP<=(RVMPEP+1) THEN LET C$=CHR$(39): GO TO 1025
1000 IF CODE C$=10 THEN LET PVCP=(PVCP+2): LET IH=0: LET IV=2: IF PVCP>(RVMPEP-1+DVMP) THEN LET C$=CHR$(38): GO TO 1025
1010 IF CODE C$=8 THEN LET PHCP=(PHCP-2): LET IH=-2: LET IV=0: IF PHCP<=(RHMPEP+1) THEN LET C$=CHR$(37): GO TO 1025
1020 IF CODE C$=9 THEN LET PHCP=(PHCP+2): LET IH=2: LET IV=0: IF PHCP>(RHMPEP-1+DHMP) THEN LET C$=CHR$(40): GO TO 1025
1022 IF CODE C$=94 THEN GO SUB 2000
1025 IF CODE C$=37 OR CODE C$=38 OR CODE C$=39 OR CODE C$=40 OR CODE C$=94 THEN RETURN
1030 LET PHCEMC=(H-RHMPEP)/2+RHMPPEMC
1040 LET PVCEMC=(V-RVMPEP)/2+RVMPEMC
1050 IF CODE C$=35 THEN LET IV=0: LET IH=0
1070 IF CODE C$=32 OR CODE C$>40 THEN LET A$(PVCEMC,PHCEMC)=C$: GO TO 1090
1080 GO TO 1140
1090 LET PVCP=PVCP+IV: LET PHCP=PHCP+IH
1100 IF PVCP<=(RVMPEP+1) THEN LET C$=CHR$(39): GO TO 1160
1110 IF PVCP>(RVMPEP-1+DVMP) THEN LET C$=CHR$(38): GO TO 1160
1120 IF PHCP<=(RHMPEP+1) THEN LET C$=CHR$(37): GO TO 1160
1130 IF PHCP>(RHMPEP-1+DHMP) THEN LET C$=CHR$(40): GO TO 1160
1140 PRINT AT V,H;A$(PVCEMC,PHCEMC)
1150 GO SUB 810
1160 IF CODE C$=37 OR CODE C$=38 OR CODE C$=39 OR CODE C$=40 THEN RETURN
1170 GO TO 950
1210 LET DV=0: LET DH=0
1220 IF CODE C$=39 AND RVMPEMC>5 THEN LET DV=-5
1230 IF CODE C$=39 AND

```

1240 IF CODE C\$=38 AND (DV	1290 IF CODE C\$=40 AND (DH	1500 REM
MC-(RVMPEMC+(DVMP	MC-(RHMPEMC+(DHMP	2000 REM GRABACION
-1)/2))>=5 THEN LET	-1)/2))<8 THEN LET D	2001 REM
DV=5	H=DHMC-(RHMPEMC+(2005 LET N(1)=DHMC: LET
1250 IF CODE C\$=38 AND (DV	DHMP-1)/2)	N(2)=DVMC
MC-(RVMPEMC+(DVMP	1310 LET RHMPEMC=RHMPE	2010 INPUT PAPER 7; BRIGHT
-1)/2))<5 THEN LET D	MC+DH: LET RVMPEMC	1; INK 1;"NOMBRE? ";N\$:
V=DVMC-(RVMPEMC+(D	=RVMPEMC+DV	IF N\$="" THEN GO TO
VMP-1)/2)	1330 IF CODE C\$=39 THEN LET	2010
1260 IF CODE C\$=37 AND	PVCP=RVMPEP+2	2011 IF LEN N\$>8 THEN PRINT
RHMPEMC>8 THEN LET	1340 IF CODE C\$=38 THEN LET	#0; PAPER 1; INK 7;
DH=-8	PVCP=DVMP+RVMPEP-1	BRIGHT 1; FLASH 1;
1270 IF CODE C\$=37 AND	1350 IF CODE C\$=37 THEN LET	"NOMBRE MUY LARGO
RHMPEMC<=8 THEN LET	PHCP=RHMPEP+2	(Max. 8)": PAUSE 0: GO TO
DH=-RHMPEMC	1360 IF CODE C\$=40 THEN LET	2010
1280 IF CODE C\$=40 AND (DH	PHCP=DHMP+RHMPEP-1	2017 PRINT #0; PAPER 7;
MC-(RHMPEMC+(DHMP	1370 GO SUB 870: RETURN	BRIGHT 1; INK 1;
		"CASSETTE/DISCO/
		MICRODRIVE? C/D/M"
		2019 LET X\$=INKEY\$: IF
		X\$="C" OR X\$="D" OR
		X\$="M" THEN GO TO 2021
		2020 GO TO 2019
		2021 IF X\$="D" OR X\$="M"
		THEN SAVE "*"m";1;
		N\$+"D" DATA N(): SAVE
		"*m";1;N\$ DATA A\$()
		2030 IF X\$="C" THEN SAVE
		N\$+"D" DATA N(): SAVE
		N\$ DATA A\$()
		2031 INPUT PAPER 7; BRIGHT
		1; INK 1;"CASSETTE/
		DISCO/MICRODR. C/D/M"
		2035 CLS : GO SUB 560:
		RETURN
		9998 STOP
		9999 SAVE "*"M";1;"CRUCI"

FE DE ERRATAS

Por error, en el anterior número de INPUT, algunas de la líneas de programa publicadas figuraban con incorrecciones. Se trata de las siguientes líneas del programa REDEFINE TUS GRAFICOS DE PANTALLA; la primera línea 300 debe suprimirse, y en su lugar escribir las líneas:

```
210 FOR C=162 TO 174
220 PLOT INK 2:W,C
```

Además, en la segunda línea 300 falta un punto y coma entre las segundas comillas y la sentencia AT.

LA MAQUINA TRAGAPERRAS

Ven con nosotros a probar suerte en Las Vegas con el juego de la máquina de frutas que te ofrece INPUT. No tengas miedo: nuestra tragaperras no te vaciará los bolsillos.

La gran mayoría de bingos y salones de juego han introducido, en lugar de las antiguas máquinas de factura mecánica, otras modernas de frutas electrónicas, con una pantalla de televisión que muestra las ruletas.

Este artículo se divide en dos partes, y trata de enseñarte cómo puedes imitar una de estas innovadoras máquinas con la ayuda de tu micro.

El juego está dotado de todos los componentes que puedes encontrar en una máquina de frutas de verdad —avance, apuesta, movimiento, etcétera— y se dan gráficos animados para simular las ruletas mecánicas.

El disponer de un programa como éste para jugar te va a salvar de que un jugador avaricioso te desplume, aunque tampoco te vas a hacer rico con él. No olvides, como siempre, guardar (SAVE) esta parte del programa que completarás con la segunda. Para hacer funcionar (RUN) el programa habrás de esperar a tenerlo completo, aunque ya puedes probar la visualización de algunos gráficos fundamentales.

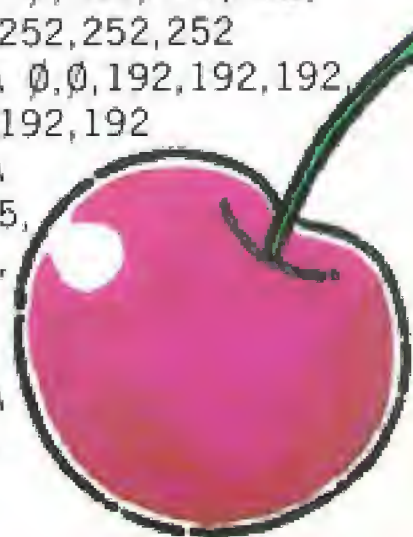
INICIALIZACION

```
40 POKE 53280,0:POKE 53281,
0:A$=" ESPERAR U":B$="N
MOMENTO
":C$="[CTRL+2][CTRL+8]
```

```
[COMM+3]
[CTRL+5][COMM.+1]
[CTRL+3]
[COMM+2]"
45 PRINT "[SHIFT+CLR/HOME][
9*CRSR ABAJO]":FOR A=1
TO 7:FOR B=1 TO 10:PRINT
"[CRSR ARRIBA]";
MID$(C$,A,1);SPC(B);
A$;
50 PRINT TAB(30-B);B$:NEXT
B:PRINT :NEXT A
55 POKE 52,48:POKE 56,
48:CLR:POKE 56334,
PEEK(56334) AND 254:POKE
1,PEEK(1)
AND 251
60 FOR A=0 TO 1023:POKE
A+12288,
PEEK(A+53248):NEXT
:POKE 1,PEEK(1)
OR 4
65 POKE 56334,PEEK
(56334) OR 1:POKE
53280,8:FOR
A=0 TO
655:
READ B:
POKE
12288
+A,B:
NEXT A
10000 DATA
255,213,
253,253,253,253,213,
255
10005 DATA 255,127,127,
127,127,127,87,255
10010 DATA 255,213,255,
213,213,215,213,255
10015 DATA 255,87,215,87,
87,255,87,255
10020 DATA 255,213,255,
253,253,255,213,255
10025 DATA 255,87,215,87,
```

■	FRUTAS MECANICAS
■	SACA EL MAXIMO PARTIDO
	DE LOS GRAFICOS
■	GRAFICOS DE FRUTAS
■	CREACION DE LAS RULETAS

```
87,215,87,255
10030 DATA 255,215,215,
213,213,255,255,255
10035 DATA 255,215,215,87,
87,215,215,255
10040 DATA 255,213,215,
213,213,255,213,255
10045 DATA 255,87,255,87,
87,215,87,255
10050 DATA 255,204,0,240,
252,204,204,240
10055 DATA 255,204,0,48,
252,204,204,252
10060 DATA 252,204,0,240,
252,204,204,240
10065 DATA 204,204,252,
240,0,204,255,0
10070 DATA 204,204,204,
204,0,204,255,0
10075 DATA 204,204,204,
204,0,204,252,0
10080 DATA 0,0,0,3,3,15,15,0
10085 DATA 48,252,252,255,
255,255,255,0
10090 DATA 0,0,0,0,0,192,
192,0
10095 DATA 15,15,3,0,0,
0,0,0
10100 DATA 255,255,255,
252,48,63,15,0
10105 DATA 192,192,0,0,0,0,
0,0
10110 DATA 0,3,15,15,15,15,
15,15
10115 DATA 0,255,243,252,
252,252,252,252
10120 DATA 0,0,192,192,192,
192,192,192
10125 DATA
15,15,
15,3,
3,0,
0,0
10130 DATA
252,
```



PROGRAMACION DE JUEGOS



```

910 PRINT PAPER 6; INK 4; AT
  1,4;" "; AT 4,4;" "
920 PRINT PAPER 6; INK 2; AT
  1,4;" "; AT 5,4;" "; INK 0;" "
930 FOR i=6 TO 14: FOR j=8
  TO 23 STEP 5: PRINT INK
  0; AT i,j;" ": NEXT j: NEXT i
940 FOR i=8 TO 23: PRINT AT
  15,i; INK 0;" ": NEXT i
950 FOR i=6 TO 21: PRINT
  PAPER 6; AT i,0;" "; AT i,24;"
  "
  NEXT i
960 FOR i=16 TO 21: PRINT
  PAPER 6; AT i,8;" "
  NEXT i
970 PRINT INK 1; PAPER 8; AT
  2,2;"GANO"; AT 2,27;"
  "GANO"; AT 3,2;" "
  "AT 3,
  27;" "
980 PRINT INK 1; PAPER 8; AT
  4,3;"10"; AT 4,28;"100"; AT
  6,3;"20"; AT 8,3;"50"; AT 7,
  25;"JACKPOT"; AT 8,25;" "
990 PRINT INK 1; PAPER 8; AT
  20,4;"PRESIONA SPACE
  PARA GIRAR"; AT 21,4;" O
  TECLAS DE PARAR O
  MOVER"
1000 PRINT AT 17,7;" 1  2
  3 "; AT 16,9;" "
1010 PRINT AT 18,7;" 4=1&2
  5=2&3 6=1&3"; AT 19,7;
  INK 7; PAPER 2;"TECLAS
  PARAR 1 a 6 "
1020 RETURN
  
```

La línea 10 salta a la subrutina de la línea 860, la cual se encarga de dibujar la máquina de frutas. Todas las instrucciones necesarias para el jugador se encuentran visualizadas en la parte frontal de la máquina.

La línea 100 establece los UDG mediante la lectura (READ) de los DATA de las líneas 20 a 90. Los UDG aparecen, una vez establecidos, en las líneas 120 a 170 como DATA prontos para dar marcha a las ruletas; a\$, b\$ y c\$ son las tres ruletas, y el orden de las frutas en cada una de ellas se determina según el orden en que están dispuestos los DATA.

VA DE RULETAS

```

180 GO SUB 1030
1030 PRINT INK 0; AT 5,0;
  A$(4): FOR X=0 TO 3
  STEP 2: PRINT INK 0; AT
  7,X; A$(4);" ": NEXT X:
  FOR X=0 TO 5 STEP 2:
  PRINT INK 0; AT 9,X;
  A$(4): NEXT X: FOR X=0
  TO 5 STEP 2: PRINT AT
  10,X; A$(1);" ": NEXT X
1040 PRINT INK 1; AT 11,0;
  C$(2); C$(2); C$(2); INK 0;
  AT 12,0; B$(1); B$(1); B$(1)
1050 PRINT INK 3; AT 5,26;
  C$(1); C$(1); C$(1); INK 4;
  AT 6,26; C$(3); C$(3);
  C$(3); INK 5; AT 9,26;
  C$(4); C$(4); C$(4)
1060 PRINT AT 7,10; A$(1); AT
  
```

```

  10,10; A$(2); AT 13,10;
  A$(3)
1070 PRINT AT 7,15; B$(1); AT
  10,15; B$(2); AT 13,15;
  B$(3)
1080 PRINT AT 7,20; C$(1); AT
  10,20; C$(2); AT 13,20;
  C$(3)
1090 PRINT AT 16,26; INK 2;
  PAPER 6;"MOVER"; AT 17,
  26;"TECLAS"; AT 15,25;
  "Q-W-E P"; AT 18,25;
  "A-S-D O"
1100 PRINT INK 0; PAPER 7; AT
  12,25;" "; AT 13,25;" ";
  AT 13,31;" "; AT 14,25;"
  "; AT 13,26; INK 2;" "
1110 PRINT PAPER 6; INK 0;
  INVERSE 1; AT 15,0;
  "TOTAL"; INVERSE 0; AT
  18,0;"p"; AT 17,1; PAPER
  7; BRIGHT 1;" "; AT 18,1;
  TOTAL
1120 PRINT #1; AT 0,0;"
  COMIENZAS CON 100
  pts."
1130 PAUSE 0
1140 PRINT #1; AT 0,0;"
1150 RETURN
  
```

La subrutina que se inicia en la línea 1030 diseña las ruletas en sus posiciones iniciales y completa algunos de los detalles de su visualización. Se le indica, además, al jugador que comienza con un dólar, y que cada vuelta cuesta diez centavos.

FRUTAS Y COCO

■	COMPLETA TU MAQUINA DE FRUTAS
■	DAR VUELTAS A LAS RULETAS
■	COMPROBACION DE LAS

■	LINEAS GANADORAS
■	AJUSTE DEL CREDITO
■	PARPADEO DE LOS AVANCES, PARADAS Y APUESTAS
■	LOS EFECTOS SONOROS

Libera la ruleta de tu tragaperras y trata de conseguir el frutero con esta segunda parte del programa. Se cobra 20 veces más que se apuesta.

¡Y a ver si en esta ocasión la diosa Fortuna se pone de tu lado y puedes, por fin, realizar tu anhelo de hacer quebrar la banca! ¡Ánimo y suerte!

En esta segunda parte del artículo que hemos dedicado a la máquina de frutas, se completa el programa y de-

jamos la máquina a punto para empezar el juego.

CONTROLENSE, SEÑORES

```
190 LET HOLD=0
200 LET TOTAL=TOTAL-10: GO
SUB 750: IF TOTAL<0
THEN GO TO 770: LET
NUDGE=0: PRINT AT 13,
26: INK 2;"
```

```
210 IF HFLAG=0 THEN LET
HOLD=0
220 FOR I=1 TO 3: FOR J=1
TO 12: BEEP .001,
60
230 IF HOLD=0 THEN PRINT AT
7,10;A$(J);AT 7,15;B$(J);
AT 7,20;C$(J);AT 10,10;
A$(J+1);AT 10,15;B$(J+1);
AT 10,20;C$(J+1);AT 13,
10;A$(J+2);AT 13,15;
```



PROGRAMACION DE JUEGOS



```

310 IF HOLD<>2 AND
    HOLD<>5 AND HOLD<>4
    THEN LET K=INT
    (RND*12): IF K=0 THEN
    LET K=1
320 IF HOLD<>3 AND
    HOLD<>5 AND HOLD<>6
    THEN LET L=INT
    (RND*12): IF L=0 THEN
    LET L=1
330 LET HOLD=0
340 PRINT AT 7,10;A$(M);AT 7,
    15;B$(K);AT 7,20;C$(L);AT
    10,10;A$(M+1);AT 10,15;
    B$(K+1);AT 10,20;
    C$(L+1);AT 13,10;
    A$(M+2);AT 13,15;
    B$(K+2);AT 13,20;
    C$(L+2)

```

La línea 190 pone a cero la variable de parada (HOLD). La rutina comprueba el valor de HOLD (determinada por la tecla pulsada por el jugador) y pone en movimiento las ruletas liberadas.

Una vez que las ruletas han dado vueltas, HOLD vuelve a valer cero en la línea 330 (los botones del hold desaparecen) y las ruletas quedan visualizadas (PRINT) en posición de reposo en la línea 340.

```

B$(J+2);AT 13,20;C$(J+2):
NEXT J:
NEXT I
240 IF HOLD=1 THEN PRINT AT
    7,15;B$(J);AT 7,20;C$(J);
    AT 10,15;B$(J+1);AT 10,
    20;C$(J+1);AT 13,15;
    B$(J+2);AT 13,20;C$(J+2):
    NEXT J:
    NEXT I
250 IF HOLD=4 THEN PRINT AT
    7,20;A$(J);AT 10,20;B$(J);
    AT 13,20;C$(J): NEXT J:
    NEXT I
260 IF HOLD=6 THEN PRINT AT
    7,15;B$(J);AT 10,15;
    B$(J+1);AT 13,15;
    B$(J+2): NEXT J:
    NEXT I
270 IF HOLD=2 THEN PRINT AT
    7,10;A$(J);AT 7,20;C$(J);

```

```

    AT 10,10;A$(J+1);AT 10,
    20;C$(J+1);AT 13,10;
    A$(J+2);AT 13,20;C$(J+2):
    NEXT J: NEXT I
280 IF HOLD=5 THEN PRINT AT
    7,10;A$(J);AT 10,10;
    A$(J+1);AT 13,10;A$(J+2):
    NEXT J:
    NEXT I
290 IF HOLD=3 THEN PRINT AT
    7,10;A$(J);AT 7,15;B$(J);
    AT 10,10;A$(J+1);AT 10,
    15;B$(J+1);AT 13,10;A$(
    J+2);AT 13,15;B$(J+2):
    NEXT J:
    NEXT I
300 IF HOLD<>1 AND
    HOLD<>4 AND HOLD<>6
    THEN LET M=INT
    (RND*12): IF M=0 THEN
    LET M=1

```

FRUTAS DEL AZAR

```

350 GO SUB 600
600 LET
    T$=A$(M)+B$(K)+C$(L)
610 LET
    M$=A$(M+1)+B$(K+1)+
    C$(L+1)
620 LET
    L$=A$(M+2)+B$(K+2)
    +C$(L+2)
630 IF M$( TO 2)=C$(1) AND
    M$( TO 2)=M$(3 TO 4) AND
    M$( TO 2)=M$(5 TO ) THEN
    LET TOTAL=TOTAL+100:
    GO SUB 750: GO SUB 760:
    GO TO 380
640 IF M$( TO 2)=C$(3) AND
    M$( TO 2)=M$(3 TO 4) AND
    M$( TO 2)=M$(5 TO ) THEN
    LET TOTAL=TOTAL+100:

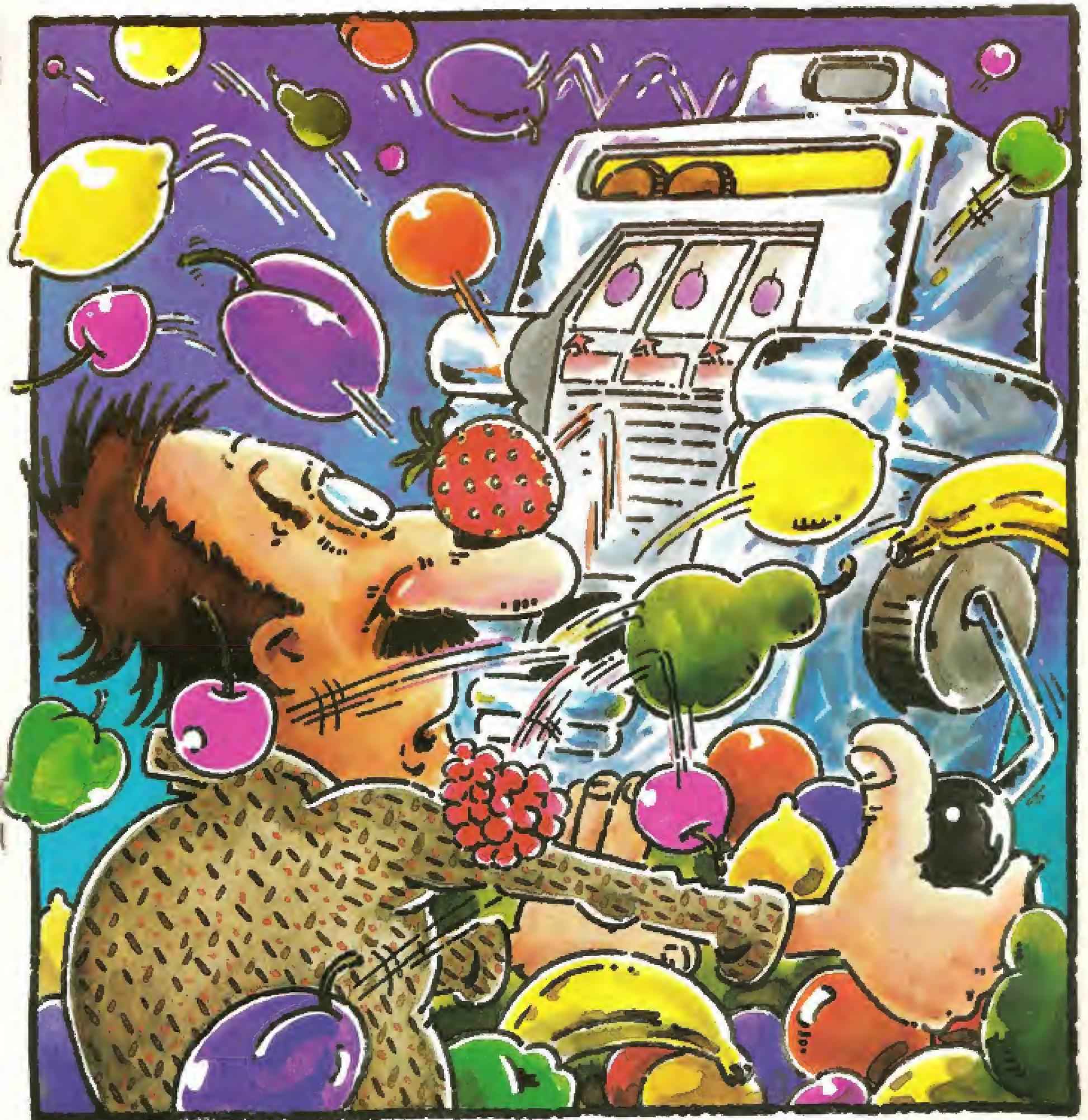
```


PROGRAMACION DE JUEGOS

GO SUB 750: GO SUB 760:
GO TO 380
650 IF M\$(TO 2)=C\$(9) AND
M\$(TO 2)=M\$(3 TO 4) AND
M\$(TO 2)=M\$(5 TO) THEN
LET TOTAL=TOTAL+50: GO
SUB 750: GO SUB 760: GO

TO 380
660 IF M\$(TO 2)=C\$(2) AND
M\$(TO 2)=M\$(3 TO 4) AND
M\$(TO 2)=M\$(5 TO) THEN
LET TOTAL=TOTAL+50: GO
SUB 750: GO SUB 760: GO
TO 380

670 IF M\$(TO 2)=C\$(6) AND
M\$(TO 2)=M\$(3 TO 4) AND
M\$(TO 2)=M\$(5 TO) THEN
LET TOTAL=TOTAL+50: GO
SUB 750: GO SUB 760: GO
TO 380
680 IF M\$(TO 2)=C\$(5) AND




```

M$( TO 2)=M$(3 TO 4) AND
M$( TO 2)=M$(5 TO ) THEN
LET TOTAL=TOTAL+50: GO
SUB 750: GO SUB 760: GO
TO 380
690 IF M$( TO 2)=C$(4) AND
M$( TO 2)=M$(3 TO 4) AND
M$( TO 2)=M$(5 TO ) THEN
LET TOTAL=TOTAL+500:
GO TO 820
700 IF M$( TO 2)=C$(5) AND
M$( TO 2)=M$(3 TO 4) AND
M$(5 TO )<>C$(5) THEN
LET TOTAL=TOTAL+20: GO
SUB 750: GO SUB 760: GO
TO 380
710 IF M$( TO 2)=C$(5) AND
M$(3 TO 4)<>C$(5) THEN
LET TOTAL=TOTAL+10: GO
SUB 750: GO SUB 760: GO
TO 380
720 IF M$( TO 2)=C$(4) AND
M$( TO 2)=M$(3 TO 4) AND
M$( TO 2)<>M$(5 TO )
THEN GO SUB 760: GO TO
380
730 IF M$( TO 2)=C$(4) AND
M$( TO 2)<>M$(3 TO 4)
THEN GO SUB 760: GO TO
380
740 IF M$(3 TO 4)=C$(4) AND
M$(3 TO 4)<>M$(5 TO )
THEN GO SUB 760: GO TO
380
750 LET DD=INT (TOTAL/100):
LET CC=TOTAL-(DD*100):
PRINT INK 2; PAPER 6;AT
17,0;"P ";AT 18,0;"P ";
PAPER 7; BRIGHT 1;AT 17,
1;" ";DD;AT 18,1;" ";CC:
RETURN
760 LET HFLAG=0: LET
HOLD=0: LET NUDGE=0:
FOR I=9 TO 19 STEP 5:
PRINT AT 16,1; INK 2;
" ": NEXT I: PRINT AT
13,26; INK 2;" ":
RETURN

```

La línea 350 salta a la subrutina que comienza en la línea 600, la cual se encarga de poner las tres filas visibles en las ruletas dentro de tres variables de

cadena, T\$, M\$ y L\$. La variable del medio, M\$, es la línea a partir de la cual se calcula el resultado en la subrutina que se inicia en la línea 630. La rutina comprueba si existen líneas ganadoras y suma las ganancias en el total del jugador.

Y EL «NUDGE» QUE RUGE

```

360 IF M<7 OR K=L OR L>2
THEN LET NUDGE=1:
PRINT BRIGHT 1; PAPER 7;
INK 2;AT 13,26;
"MOVER"
370 LET HFLAG=INT (RND+.5):
IF HFLAG=1 THEN FOR I=9
TO 19 STEP 5: PRINT AT
16,I; INK 6; BRIGHT 1;
"ALTO":
NEXT I
380 LET I$=INKEY$: IF I$=""
THEN GO TO 380
390 IF I$=" " THEN FOR I=9
TO 19 STEP 5: PRINT INK
2;AT 16,1;" ": NEXT I:
GO TO 200
400 IF I$="E" AND NUDGE=1
THEN GO SUB 520: LET
NUDGE=0: PRINT AT 13,
26; INK 2;" ": BEEP .1,
30: GO SUB 600: LET
RN=INT (RND*10): IF INT
(RN/2)=RN/2 AND
HFLAG<>1 THEN LET
NUDGE=1: PRINT AT 13,
26; INK 7; BRIGHT 1;
"MOVER": GO TO
380
410 IF I$="Q" AND NUDGE=1
THEN GO SUB 480: LET
NUDGE=0: PRINT AT 13,
26; INK 2;" ": BEEP .1,
30: GO SUB 600: LET
RN=INT (RND*10): IF INT
(RN/2)=RN/2 AND RN<3
THEN LET NUDGE=1:
PRINT AT 13,26; INK 7;
BRIGHT 1;"MOVER": GO TO
380
420 IF I$="W" AND NUDGE=1
THEN GO SUB 500: LET
NUDGE=0: PRINT AT 13,

```

```

26; INK 2;" ": BEEP .1,
30: GO SUB 600: GO TO
380
430 IF I$="D" AND NUDGE=1
THEN GO SUB 540: LET
NUDGE=0: PRINT AT 13,
26; INK 2;" ": BEEP .1,
30: GO SUB 600: LET

```



PROGRAMACION DE JUEGOS

```

RN=INT (RND*10): IF INT
(RN/2)<>RN/2 THEN LET
NUDGE=1: PRINT AT 13,
26; INK 7; BRIGHT 1;
"MOVER": GO TO
380
440 IF I$="S" AND NUDGE=1
THEN GO SUB 580: LET

```

```

NUDGE=0: PRINT AT 13,
26; INK 2;"    ": BEEP .1,
30: GO SUB 600: LET
RN=INT (RND*10): IF INT
(RN/2)=RN/2 THEN LET
NUDGE=1: PRINT AT 13,
26; INK 7; BRIGHT 1;
"MOVER": GO TO

```

```

380
450 IF I$="A" AND NUDGE=1
THEN GO SUB 560: LET
NUDGE=0: PRINT AT 13,
26; INK 2;"    ": BEEP .1,
30: GO SUB 600: LET
RN=INT (RND*10): IF INT
(RN/2)<>RN/2 AND RN>6
THEN LET NUDGE=1:
PRINT AT 13,26; INK 7;
BRIGHT 1;"MOVER": GO TO
380
460 IF HFLAG=1 AND I$="1"
OR I$="2" OR I$="3" OR
I$="4" OR I$="5" OR
I$="6" THEN LET
HOLD=VAL I$: FOR I=9 TO
19 STEP 5: PRINT AT 16,I;
INK 2;"    ": NEXT I: GO
TO 200
470 GO TO 380
480 LET M=M+1: IF M>12
THEN LET M=
M-12
490 PRINT AT 7,10;A$(M);AT
10,10;A$(M+1);AT 13,10;
A$(M+2);
RETURN
500 LET K=K+1: IF K>12
THEN LET K=
K-12
510 PRINT AT 7,15;B$(K);AT
10,15;B$(K+1);AT 13,15;
B$(K+2);
RETURN
520 LET L=L+1: IF L>12 THEN
LET L=L-12
530 PRINT AT 7,20;C$(L);AT
10,20;C$(L+1);AT 13,20;
B$(L+2);
RETURN
540 LET L=L-1: IF L<1 THEN
LET L=L+12
550 PRINT AT 7,20;C$(L);AT
10,20;C$(L+1);AT 13,20;
B$(L+2);
RETURN
560 LET M=M-1: IF M<1
THEN LET M=
M+12
570 PRINT AT 7,10;A$(M);AT
10,10;A$(M+1);AT 13,

```




```

10;A$(M+2):
RETURN
580 LET K=K-1: IF K<1 THEN
LET K=K+12
590 PRINT AT 7,15;B$(K);AT
10,15;B$(K+1);AT 13,15;
B$(K+2):
RETURN
    
```

La rutina del NUDGE, encargada del movimiento, es similar a la rutina del *hold*.

Las ruletas pueden moverse hacia arriba o hacia abajo, según lo que haya decidido entrar el jugador por el teclado. Las instrucciones sobre la tecla que se ha de pulsar se visualizan en la pantalla. A cada *nudge* se genera un número aleatorio que determinará si se puede ofrecer o no un nuevo *nudge*.

JACKPOT: UN CUERNO DE LA ABUNDANCIA

```

770 CLS : PRINT AT 10,0;"
      FINAL DEL JUEGO
      PERDISTE TODO EL
      DINERO": BEEP 1,
      -20
780 PRINT "" QUIERES JUGAR
      OTRA VEZ? PRESIONA
      ""S"" O ""N""
790 IF INKEY$="" THEN GO TO
      790
800 LET I$=INKEY$: IF I$="S"
      OR I$="s" THEN RUN
810 STOP
820 CLS : PRINT AT 10,0;"
      FELICIDADES GANASTE
      EL JACKPOT": PRINT
      "ERES EL MAS RICO CON
      5000 PTS": FOR J=1 TO 3:
      FOR I=1 TO 10: BEEP .01,
      5*I: NEXT I:
      NEXT J
830 GO TO 780
    
```

Estas rutinas son muy sencillas. La rutina que pregunta si quieres intentarlo de nuevo, contenida en las líneas 770 a 810, es llamada cuando al jugador ya no le queda dinero para apostar.

La rutina del *jackpot*, frutero de premio (líneas 820 y 830), da una



buena noticia al jugador y le suma 50 dólares a su dinero total, antes de concluir la jugada: la banca ha quebrado. Al jugador se le concede ahora una nueva oportunidad.

64 COLUMNAS EN BASIC

Una de las limitaciones del Spectrum, a la hora de utilizarlo para aplicaciones tipo procesadores de texto, bases de datos, etc., reside en que sólo dispone de 32 columnas, número claramente insuficiente para este tipo de necesidades.

La siguiente rutina permite ubicar dos caracteres en el lugar que antes empleaba uno solo. O sea, sus dimensiones, con la nueva rutina, serán de 8×4 pixels, cuando antes eran de 8×8 .

Para conseguir esto, lo más cómodo es imprimir un carácter en los cuatro primeros pixels y el otro carácter en los cuatro últimos. Para ello tendremos que hacer dos nuevos juegos de caracteres; el primero con las letras en los cuatro últimos pixels, y el segundo en los cuatro primeros (fig. 2a y 2b). Cada juego consta de 96 caracteres (códigos ASCII del 32 al 127, ambos inclusive) y cuenta con una longitud de 768 bytes (96×8), almacenados a partir de la dirección 63232 el primero y 64000 el segundo.

Los pasos a seguir para imprimir dos caracteres en uno son los siguientes: primero se imprime el primer carácter (fig. 2a); luego, en la misma línea y con OVER 1, para que no se borre lo anteriormente escrito, se imprime el otro carácter (fig. 2b), y ya que los dos caracteres no coinciden entre sí, quedarán los dos impresos en un mismo carácter (fig. 2c).

La función del programa 1 es cargar en memoria, a partir de la dirección 63232, los dos nuevos juegos de caracteres, encargándose de grabar y verificar. Si se produce algún error al introducir los datos, el programa mostrará cuál es la línea errónea.

```

1 REM PROGRAMA 1
  NUEVO JUEGO DE CARACTERES
10 CLEAR 60000: LET n=63232: RESTORE
20 FOR i=1000 TO 1095
30 LET cont=0
40 READ a$
50 FOR j=1 TO LEN a$-1 STEP 2
60 LET d=16*(CODE a$(j)-48-7*(a$(j)>"9"))+CODE a$(j+1)-48-7*(a$(j+1)>"9")
70 POKE n,d: LET cont=cont+d: LET n=n+1: NEXT j
80 READ a
90 IF a<>cont THEN GO TO 200
95 PRINT AT 0,0;"LINEA: ";i
100 NEXT i
110 PRINT AT 0,0;"OK. TODO CORRECTO";AT 1,0;"Pulsa una tecla para grabar CHR$ ": SAVE "
    NEW CHR$ SET"CODE 63232,1536
120 CLS : PRINT AT 0,0;"PULSA UNA TECLA PARA VERIFICAR.": VERIFY " NEW CHR$
    SET"CODE 63232,1536
130 PRINT AT 4,0;"GRABACION CORRECTA."
140 STOP
200 PRINT "ERROR DE DATA EN LA LINEA ";i: PRINT "REVISA LAS DATAS": STOP
1000 DATA "0000000000000000000000002020202000200",10
1001 DATA "000A0A0000000000000000002000202020200",30
1002 DATA "0E000C0A0A0A0A000E000A0E0E0A0A00",138
1003 DATA "06000C02060A06000004080000000000",54
1004 DATA "00020404040402000004020202020400",36
1005 DATA "00000A040E040A00000004040E040400",72
1006 DATA "0000000000202040000000000000E000000",22

```



```

1007 DATA "0000000000006060000002020404080800",40
1008 DATA "00040A0E0A0A040000040C0404040E00",94
1009 DATA "00040A0204080E000000C020402020C00",76
1010 DATA "0008080A0E0202000000E080C02020C00",94
1011 DATA "0006080C0A0A04000000E020404080800",90
1012 DATA "00040A040A0A040000040A0A06020C00",86
1013 DATA "00000400000400000000000400040800",24
1014 DATA "000002040804020000000000E000E0000",48
1015 DATA "00000804020408000000C020204040004",54
1016 DATA "040004040808060000040A0A0E0A0A00",92
1017 DATA "000C0A0C0A0A0C000006080808080600",110
1018 DATA "000C0A0A0A0A0C00000E080C08080E00",128
1019 DATA "000E080C08080800000608080A0A0600",106
1020 DATA "000A0A0E0A0A0A00000E040404040E00",108
1021 DATA "000602020A0A0400000A0A0C0A0A0A00",96
1022 DATA "0008080808080E0000090F0909090900",114
1023 DATA "000A0E0E0A0A0A0000040A0A0A0A0400",116
1024 DATA "000C0A0A0C08080000040A0A0A0E0600",114
1025 DATA "000C0A0A0C0A0A000006080402020C00",98
1026 DATA "000E040404040400000A0A0A0A0A0400",88
1027 DATA "000A0A0A0A0404000009090D0D060600",104
1028 DATA "000A0A04040A0A00000A0A0A04040400",90
1029 DATA "000E020404080E000006040404040600",74
1030 DATA "00080804040202000006020202020600",48
1031 DATA "00040E0404040400000000000000000F",49
1032 DATA "0003040604040F0000000C02060A0600",72
1033 DATA "0008080C0A0A0C000000060808080600",96
1034 DATA "000202060A0A06000000040A0C080600",76

```



1035 DATA "0006080C0808080000000060A06020C00",86
 1036 DATA "0008080C0A0A0A000000004000C040E00",92
 1037 DATA "00020002020A0400000000A0A0C0A0A00",72
 1038 DATA "000008080808060000000090F09090900",89
 1039 DATA "00000C0A0A0A0A0000000040A0A0A0400",90
 1040 DATA "00000C0A0A0C080800000060A0A060202",96
 1041 DATA "00000A0C0808080000000060804020C00",78
 1042 DATA "00040E0404040200000000A0A0A0A0400",76
 1043 DATA "00000A0A0A04040000000090D0D060600",85
 1044 DATA "00000A0A040A0A00000000A0A0602020C",86
 1045 DATA "00000E0204080E0000600040A0C080600",88
 1046 DATA "000404040404040000204000C04040E00",64
 1047 DATA "0600040A0A0A040000204000A0A0A0400",84
 1048 DATA "000000000000000000000040404040004000",320
 1049 DATA "00A0A0000000000000000040004040404000",640
 1050 DATA "00E000C0A0A0A000E000A0E0E0A0A000",2048
 1051 DATA "6000C02060A0E000000408000000000000",992
 1052 DATA "00204040404020000080404040408000",832
 1053 DATA "0000A040E040A000000004040E0404000",1152
 1054 DATA "00000000202040000000000000E00000000",352
 1055 DATA "00000000000C0C00000020204040808000",832
 1056 DATA "0040A0E0A0A040000040C0404040E000",1504
 1057 DATA "0040A0204080E000000C020402020C000",1216
 1058 DATA "008080A0E020200000E080C020A04000",1504
 1059 DATA "006080C0A0A0400000E0204040808000",1440
 1060 DATA "0040A040A0A040000040A0A06020C000",1376
 1061 DATA "000080000080000000000004000408000",512
 1062 DATA "0000204080402000000000E000E00000",768
 1063 DATA "0000804020408000000C0202040400040",864
 1064 DATA "40004040808060000040A0A0E0A0A000",1472
 1065 DATA "00C0A0C0A0A0C0000060808080806000",1760
 1066 DATA "00C0A0A0A0A0C00000E080C08080E000",2048
 1067 DATA "00E080C08080800000608080A0A06000",1696
 1068 DATA "00A0A0E0A0A0A00000E040404040E000",1728
 1069 DATA "00602020A0A0400000A0A0C0A0A0A000",1536
 1070 DATA "008080808080E0000090F09090909000",1824
 1071 DATA "00A0E0E0A0A0A0000040A0A0A0A04000",1856
 1072 DATA "00C0A0A0C08080000040A0A0A0E06000",1824
 1073 DATA "00C0A0A0C0A0A000006080402020C000",1568
 1074 DATA "00E040404040400000A0A0A0A0A04000",1408
 1075 DATA "00A0A0A0A0404000009090D0D0606000",1664
 1076 DATA "00A0A04040A0A00000A0A0A040404000",1440
 1077 DATA "00E020404080E0000060404040406000",1184
 1078 DATA "008080404020200000C040404040C000",1088
 1079 DATA "0040E0404040400000000000000000F0",784
 1080 DATA "003040604040F0000000C02060A06000",1152
 1081 DATA "008080C0A0A0C0000000608080806000",1536
 1082 DATA "00202060A0A06000000040A0E0806000",1248
 1083 DATA "00006080C0808000000060A06020C000",1248
 1084 DATA "008080C0A0A0A00000004000C040E000",1472
 1085 DATA "0020002020A040000000A0A0C0A0A000",1152


```

1086 DATA "0000808080806000000090F090909000",1424
1087 DATA "0000C0A0A0A0A000000040A0A0A04000",1440
1088 DATA "0000C0A0A0C08080000060A0A0602020",1536
1089 DATA "0000A0C080808000000060804020C000",1248
1090 DATA "0040E040404020000000A0A0A0A04000",1216
1091 DATA "0000A0A0A0404000000090D0D0606000",1360
1092 DATA "0000A0A040A0A0000000A0A0602020C0",1376
1093 DATA "0000E0204080E000600040A0C0806000",1408
1094 DATA "0040404040404000204000C04040E000",1024
1095 DATA "600040A0A0A04000204000A0A0A04000",1344

```

El programa 2 se encarga de imprimir los caracteres; esta rutina es la que tendrás que poner en tus programas si quieres utilizar las 64 columnas. Para llamar a esta rutina hacer GO SUB 9000.

```

9000 REM PROGRAMA 2
      P$: Texto
      x: Línea del texto
      y: Columna del texto
9010 IF LEN p$/2<>INT (LEN p$/2) THEN LET p$=p$+" "
9020 OVER 1: FOR f=1 TO LEN p$ STEP 2
9030 POKE 23607,249: PRINT AT x,y;p$(f): POKE 23607,246: PRINT AT x,y;p$(f+1)
9040 LET y=y+1
9050 IF y>31 THEN LET y=0: LET x=x+1
9060 NEXT f: OVER 0
9070 POKE 23607,60
9080 RETURN

```

Hemos construido el nuevo juego de caracteres, incluyendo los caracteres hispánicos, y para que puedas realizar tú mismo los caracteres a tu gusto, también incluimos un programa creador de caracteres (Programa 3). El programa es muy sencillo, y con él podrás crear caracteres, grabarlos y cargarlos en cinta.

```

20 DEF FN a(a$)=(VAL a$(1))*2^3+(VAL a$(2))*2^2+(VAL a$(3))*2^1+(VAL a$(4))*2^0
30 GO SUB 2000
40 GO SUB 3000
100 DIM I$(8,4): FOR x=1 TO 8: LET I$(x)="0000": NEXT x
110 DIM I(8)
120 GO TO 1500
140 REM CREAM CARACTERES
150 PAPER 4: BORDER 3: INK 0: CLS : POKE 23607,60
155 GO SUB 1000
160 DRAW 255,0: DRAW 0,175: DRAW -255,0: DRAW 0,-175
165 PRINT AT 13,2: PAPER 5;" ";AT 20,2;" "; FOR f=13 TO 20: PRINT AT f,1;
      PAPER 5;" ";AT f,14;" "; NEXT f: PRINT PAPER 6;AT 14,2;"5*IZQUIERDA ";AT 15,2;
      "6*ABAJO ";AT 16,2;"7*ARRIBA ";AT 17,2;"8*DERECHA ";AT 18,2;
      "9*BORRAR ";AT 19,2;"0*PINTAR "
170 PRINT PAPER 5;AT 18,18;"ESCRITO POR";AT 19,18;"A. BURGALETA": PRINT PAPER 2;AT
      17,18;" ";AT 20,18;" "; FOR f=17 TO 20: PRINT AT f,17; PAPER
      2;" ";AT f,30;" "; NEXT f
173 PRINT PAPER 1; INK 6;AT 14,17;"INPUT SINCLAIR"
175 PLOT 72,143: DRAW 183,0: PLOT 72,143: DRAW 0,32: PRINT PAPER 2; INK 7;AT 1,10;
      "CREADOR DE CARACTERES";AT 2,10;" 64 CARACTERES/LINEA "
180 PRINT AT 6,11; PAPER 3;"1# BORRAR REJILLA ";AT 7,11;"2# VER CARACTER ";AT
      8,11;"3# GUARDAR CARACTER";AT 9,11;"4# MENU PRINCIPAL "

```



```

185 PRINT AT 5,11; PAPER 2;" ";AT 10,11;" ";FOR f=5 TO
10: PRINT AT f,10; PAPER 2;" ";AT f,30;" ";NEXT f
190 FOR f=1 TO 8: LET I$(f)="0000": NEXT f
200 PAPER 7: INK 0
210 LET x=1: LET y=1: LET x0=x: LET y0=y
230 IF INKEY$="8" AND y<4 THEN LET y0=y: LET x0=x: LET y=y+1: GO TO 300
240 IF INKEY$="5" AND y>1 THEN LET y0=y: LET x0=x: LET y=y-1: GO TO 300
250 IF INKEY$="6" AND x<8 THEN LET x0=x: LET y0=y: LET x=x+1: GO TO 300
260 IF INKEY$="7" AND x>1 THEN LET x0=x: LET y0=y: LET x=x-1: GO TO 300
270 IF INKEY$="0" THEN GO TO 400
280 IF INKEY$="9" THEN GO TO 450
285 IF INKEY$="1" THEN GO SUB 1010: FOR f=1 TO 8: LET I$(f)="0000": NEXT f
286 IF INKEY$="3" THEN GO TO 500
287 IF INKEY$="2" THEN GO TO 600
290 IF INKEY$="4" THEN GO TO 1500
299 GO TO 220
310 LET p=10*(I$(x0,y0)="0")
315 GO TO 320+p
320 PRINT AT x0+2,y0+2;" ": PRINT AT x+2,y+2;"*": GO TO 220
330 PRINT AT x0+2,y0+2;"A": PRINT AT x+2,y+2;"*": GO TO 220
400 LET I$(x,y)="1": BEEP .05,0: GO TO 220
450 LET I$(x,y)="0": BEEP .05,-10
460 GO TO 220
500 REM GUARDAR UN CHARACTER
505 BORDER 2: BEEP .05,10
506 IF I$(x,y)="0" THEN LET r$="A"
507 IF I$(x,y)="1" THEN LET r$=" "
508 PRINT AT x+2,y+2;r$
510 INPUT "EN QUE CHARACTER? ";c$
520 LET c0=CODE c$*8+62976: LET c1=CODE c$*8+63744
530 FOR f=1 TO 8: LET I(f)=FN a(1$(f)): NEXT f
540 FOR f=0 TO 7: POKE f+c0,I(f+1): NEXT f: FOR f=0 TO 7: POKE f+c1,I(f+1)*16: NEXT f
545 BEEP .05,10: BORDER 3: INPUT ""
550 GO TO 220
600 REM VER UN CHARACTER
605 BORDER 2: BEEP .05,10
610 INPUT "QUE CHARACTER? ";c$
620 POKE 23607,249: PRINT AT 11,1; PAPER 4; INK 4;c$
625 POKE 23607,60
630 FOR f=1 TO 8: LET I$(f)="0000": NEXT f
640 FOR f=1 TO 8: FOR g=1 TO 4
650 IF POINT (7+g,88-f)=1 THEN LET I$(f,g)="1"
660 NEXT g: NEXT f
670 GO SUB 900
675 BEEP .05,10: BORDER 3: INPUT ""
680 GO TO 220
900 REM DIBUJAR CHARACTER EN REJILLA
910 FOR f=1 TO 8: FOR g=1 TO 4
920 IF I$(f,g)="1" THEN PRINT AT f+2,g+2; PAPER 0;" "
930 NEXT g: NEXT f
940 RETURN

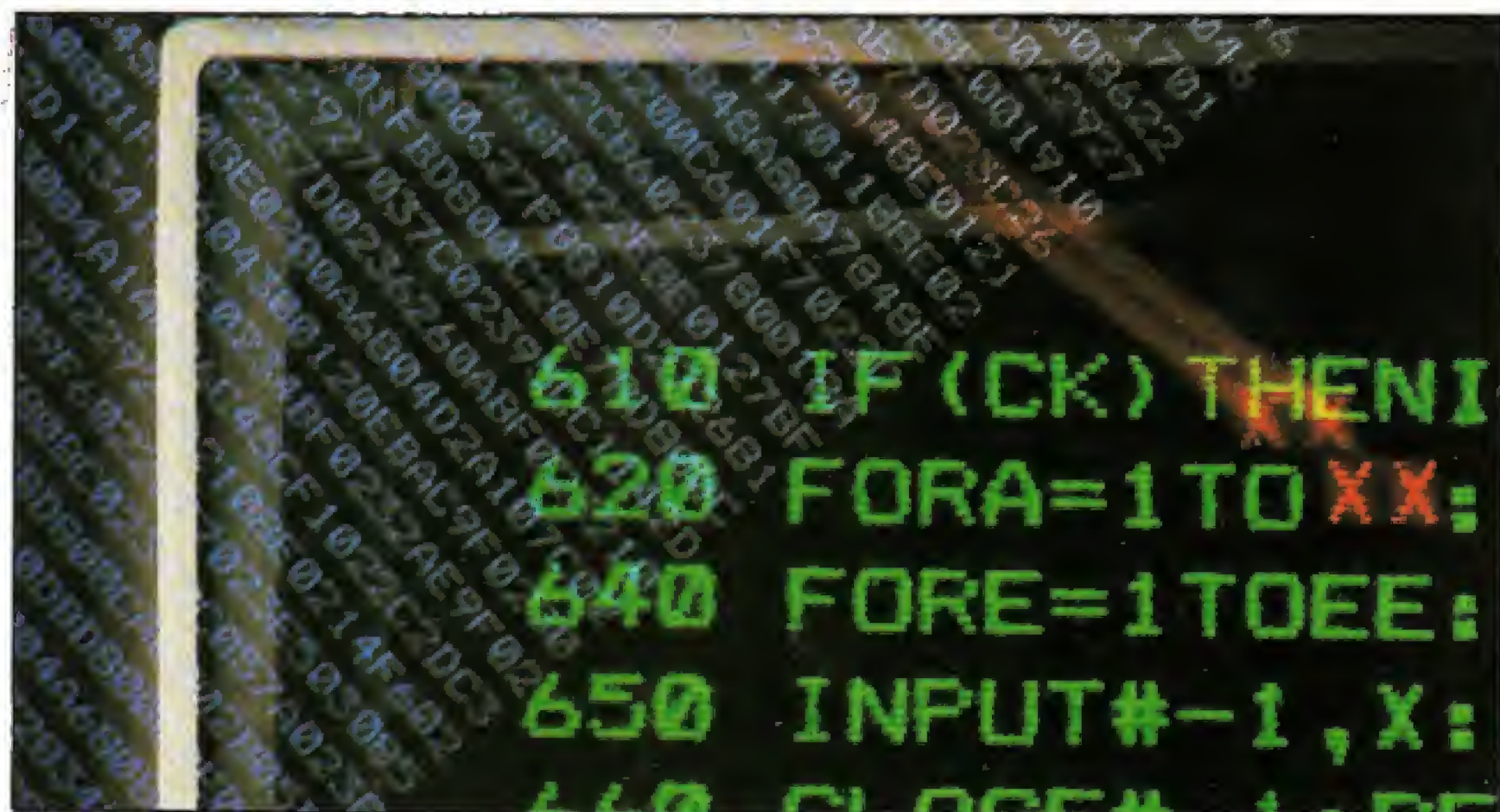
```



```

1000 REM DIBUJAR REJILLA
1010 FOR f=3 TO 10: PRINT AT f,3; PAPER 7;"AAAA": NEXT f
1015 PRINT AT 2,2; INK 3;" ";AT 11,2;" " : FOR f=3 TO 10: PRINT AT f,2; INK 3;" ";AT
    f,7;" " : NEXT f
1030 RETURN
1500 REM MENU PRINCIPAL
1510 PAPER 4: INK 0: BORDER 4: CLS
1520 PRINT AT 0,0; INVERSE 1;"CREADOR DE CARACTERES (64 CAR.)"
1530 PRINT AT 3,9;"MENU PRINCIPAL";AT 4,9;"
1540 PRINT AT 8,3;"1-CREAR CARACTERES";AT 10,3;"2-VER TODOS LOS CARACTERES";AT
    12,3;"3-GRABAR CARACTERES";AT 14,3;"4-CARGAR CARACTERES"
1550 PRINT AT 20,0;"Pulsa opcion"; FLASH 1;" "
1570 IF INKEY$="1" THEN GO TO 150
1580 IF INKEY$="2" THEN GO TO 1700
1590 IF INKEY$="3" THEN GO TO 1800
1600 IF INKEY$="4" THEN GO TO 1900
1610 GO TO 1570
1700 REM VER TODOS LOS CARAC.
1710 PAPER 5: INK 1: BORDER 5: CLS
1715 POKE 23607,60
1720 FOR f=0 TO 11 STEP 2: FOR g=1 TO 16: PRINT AT f,g*2-2; INVERSE 1;CHR$ (32+(f/
    2)*16+g-1);" "; NEXT g: NEXT f
1730 POKE 23607,249: FOR f=1 TO 12 STEP 2: FOR g=1 TO 16: PRINT AT f,g*2-2;CHR$
    (32+((f-1)/2)*16+g-1): NEXT g: NEXT f
1740 POKE 23607,60
1750 PRINT AT 19,0; FLASH 1;"PULSA UNA TECLA PARA IR AL MENU."
1760 IF INKEY$("<>") THEN GO TO 1500
1780 GO TO 1760
1800 REM GRABAR CARACTERES

```




```

1810 CLS : INPUT "NOMBRE? ";n$
1820 IF LEN n$>10 THEN LET n$=n$( TO 10)
1830 SAVE n$CODE 63232,1536
1840 PRINT #0;AT 1,0;"Pulsa una tecla para VERIFICAR"
1850 VERIFY ""CODE 63232,1536
1860 GO TO 1500
1900 REM CARGAR CARACTERES
1910 INPUT "NOMBRE? ";n$
1920 IF LEN n$>10 THEN LET n$=n$( TO 10)
1930 CLS : PRINT #0;AT 1,0;"Pulsa una tecla para CARGAR"
1940 LOAD ""CODE 63232,1536
1950 GO TO 1500
2000 REM GDU
2010 POKE USR "a",255
2020 FOR x=USR "a"+1 TO USR "a"+6: POKE x,129: NEXT x
2030 POKE USR "a"+7,255
2040 RETURN
3000 REM PASAR CARACTERES ROM A CARACTERES RAM 63232 TO 64768
3010 CLS : PRINT AT 11,8; FLASH 1;"ESPERA UN POCO"
3030 FOR x=63232 TO 63999: POKE x,PEEK (x-47616): NEXT x
3040 FOR x=64000 TO 64767: POKE x,PEEK (x-48384): NEXT x
3045 CLS
3050 RETURN

```

Al poner en marcha el programa, aparecerá en la pantalla este menú:

1. CREAR CARACTERES
2. VER TODOS LOS CARACTERES
3. GRABAR CARACTERES
4. CARGAR CARACTERES

Con la opción 1 verás en la pantalla una rejilla de 8 por 4 pixels (o cuadros en esta ocasión), las teclas de manejo y otro menú con otras cuatro opciones. En la rejilla verás un cursor (un asterisco), que podrás moverlo con las teclas del cursor (5, 6, 7 y 8). Para fijar o dibujar un cuadro en la rejilla tendrás que apretar la tecla 0; para borrarlo deberás pulsar la tecla 9. Como habrás leído antes, aquí aparece otro menú, que te ofrece las siguientes opciones:

1. BORRAR REJILLA
2. VER CARÁCTER
3. GUARDAR CARÁCTER
4. MENÚ PRINCIPAL

Con la opción 1, naturalmente, se borrará la rejilla, sin más. Con la opción 2 lo primero que hará es preguntar qué carácter quieres ver, y después lo dibujará en la rejilla. Con la 3 se guarda el carácter que esté dibujado en la rejilla, en el nuevo juego de caracteres. Y la opción 4 nos hace volver al menú principal.

Con la opción 2 podrás contemplar todos los caracteres (sólo verás un juego de éstos, ya que el otro es igual pero con el carácter dibujado en los cuatro últimos pixels).

Con las opciones 3 y 4 podrás grabar y cargar los nuevos juegos de caracteres creados por ti. Ahora ya podrás realizar cosas más serias con tu Spectrum. Y no sólo jugar, aunque las 64 columnas también lo hacen atractivo para crear juegos de aventuras con mucho texto.

Que lo disfrutes.

Asier Burgaleta

CHIP

¡JO! VAYA
DÍA LLEVO
HOY...

NUEVE
HORAS SIN
PARAR... "KNIGHTMARE,
CAMELOT WARRIORS,
COMANDO..."

SE HAN
JUNTADO TODA
LA PANDA Y CADA
UNO LLEVABA DOS
CARTUCHOS POR
LO MENOS

EL
"SOCCER"
ME LO HAN
INPUTERIZADO
20 VECES...

TENGO LOS
CIRCUITOS QUE
PARECEN EL DE LE
MANS DESPUÉS
DE CORRER LAS
24 HORAS

...

Y EL
CPU NO VEAS
COMO DEBE
ESTAR...

LE DEBE
SALIR EL SILICIO
HASTA POR LA
IMPRESORA

PERO LO
PEOR HA
VENIDO
DESPUÉS...

...CUANDO
UNO DE ELLOS
HA IDO A
POR COMIDA...

HA SIDO
TERRIBLE...
¿ADIVINAIS QUÉ
SE LE HA OCURRIDO
TRAER?

CHIP!
PATATAS
CHIP!

...Y AHÍ
SÍ QUE ME
DESTROZARON...

LAS PATATAS
CHIP SON MI
DEBILIDAD

ANO LAS
PATATAS
CHIP

SIGUEN...

BETA BASIC: EL SUPER BASIC

Seguramente tu caso será como el de la mayoría de los esforzados usuarios del Spectrum, ya pasas de tanto juego y programa repetitivo y empiezas a dominar el ordenador en profundidad. Tu interés se centra en ese otro grupo de programas llamados de utilidades, entre los que se encuentra este magnífico Beta Basic.

El citado programa se puede considerar como uno de los esenciales para tu biblioteca del Spectrum. El Beta Basic es un amplificador del BASIC de tu ordenador que añade más de 30 nuevos comandos y 20 funciones a las ya presentes. Básicamente se puede decir que es un programa en código máquina de unos 9.3 K (la versión 1.8) que una vez cargado en memoria ofrece la posibilidad de utilizar las nuevas facilidades.

En líneas generales las ventajas que aporta el programa son:

- A nivel de gráficos: Posibilidades de rellenar de color la figura deseada, movimiento de la pantalla en todas direcciones, facilidad para posicionar texto con precisión de un pixel...
- Se ha potenciado mucho el editor. El movimiento del cursor en una línea ya no sólo es horizontal. Ahora también nos podemos desplazar hacia arriba y abajo (imagínate esa «monstruosa» línea de DATAs en la que tenemos que modificar el último número de la lista, con el desplazamiento vertical del cursor podemos ahorrar mucho tiempo). Se mejora en general el tema de la edición de líneas.
- Hay que destacar la posibilidad de programación mediante procedimientos, como en PASCAL. Se han añadido también comandos utilizados por otros ordenadores, como es el caso de RENUM, AUTO, DELETE, etc... Se introduce una mayor flexibilidad a la hora de crear nuestros programas ofreciendo más facilidades de estructuración (ON x GOTO, DO UNTIL,

etc...). Existe la posibilidad del tratamiento de errores de una manera sencilla.

- Además de lo apuntado anteriormente, se abre ante nosotros un amplio abanico de posibilidades que versan sobre los más variados temas: Facilidad para fundir dos líneas en una sola, generación automática de líneas de DATAs a partir de los datos en memoria, conversores decimal-hexadecimal-binario, información sobre la cantidad de memoria disponible, y un etcétera que se extiende hasta acabar los 30 comandos y 20 funciones.

El objetivo de este artículo es indicarte los puntos más interesantes y las posibles desventajas del programa.

El creador del programa, la firma inglesa Betasoft, lleva realizadas ya tres versiones, cada una de ellas mejorando la anterior y consiguiendo una total compatibilidad ascendente de los programas realizados con cualquiera de los tres Beta Basic. La más antigua es la versión 1.0. En España quizá la más difundida sea la 1.8 que es la comercializada por la casa Ventamatic.

Ya en 1985 salió al mercado inglés la última realización, el Beta Basic 3.0 que es el más avanzado de los tres. En realidad la parte principal del programa es muy semejante en todos. Cuando cargamos el Beta Basic, aparte del bloque en código máquina se generan 3 líneas BASIC. Si hacemos la carga con MERGE, la ejecución se detiene y podemos observarlas. La primera de ellas (la 0) contiene la definición de las nuevas funciones en código máquina. Las líneas 1 y 2 nos ofrecen la posibilidad de pasar el Beta Basic a Microdrive o Discovery I, facilidad que se ha incluido para conseguir una carga rápida, ya que siempre que utilicemos algún programa o rutina que haga uso de las nuevas funciones y comandos, necesitamos previamente cargar el Beta Basic.

■	PROGRAMA DE UTILIDADES
■	BETA BASIC 1.8:
	EL MAS POPULAR
■	COMANDOS
■	FUNCIONES

Si no utilizamos esta opción cargaremos el programa normalmente con LORD, lo cual hace que se ejecute el Beta Basic haciendo uso de uno de sus comandos para borrar las líneas 1 y 2 (DELETE 1 TO 2). La línea 0 es necesaria para el funcionamiento del programa y permanece sin ser borrada, aunque no aparece en el listado a no ser que sea la única. Además de contener la definición de las nuevas funciones, la línea 0 es esencial para conseguir la perfecta adaptación de los programas realizados con cualquiera de las tres versiones. Por ejemplo, si trabajamos con el Beta Basic 1.8 grabaremos nuestros programas con la línea 0 para que cuando los carguemos de nuevo (con lo que destruimos cualquiera línea de programa) sigamos conservando la necesaria línea 0. (Recuerda que todo el Beta Basic está en código máquina protegido por el RAMTOP a excepción de la línea 0 en BASIC). Si más tarde conseguimos la versión 3.0 podremos adaptar nuestros programas hechos con la versión 1.8 con sólo cambiar al principio de cada programa o rutina la antigua línea 0 por la nueva del 3.0. Hay 2 maneras de utilizar el Beta Basic:

- La manera normal para la que fue concebido el programa. Tras estudiar el manual de instrucciones, lo cual lleva un cierto tiempo, nos acostumbramos a su uso y empezamos a sacar el jugo a este potente instrumento de programación. Nuestros listados tienen cada vez más de esos comandos inteligibles para nuestro Spectrum sin el uso del Beta Basic. El manejo de las útiles funciones se hace imprescindible. Acabamos sacando un cuadro resumen de la posición de los nuevos comandos y lo pegamos junto a nuestro teclado. Con el relativo esfuerzo que impone todo lo nuevo, hemos conseguido agrandar la potencia del BASIC de nuestro ordenador.

– Conocemos el **Beta Basic** pero somos programadores que utilizamos gran parte de la memoria del ordenador. Nos parece excesivo sacrificar casi 10 K con un programa que ha de residir permanentemente en memoria. A pesar de esto podemos sacar un gran partido al **Beta Basic**. Imagínate que quieres poner un programa en código máquina de unos 2 K en forma de DATAs. Después de imprimir en papel los datos contenidos en memoria tendrás que teclear más de ¡¡¡2000 números!!!. Con el **Beta Basic** utilizando el comando **KEYIN** generarás directamente las líneas de DATAs y tu único trabajo consistirá en grabar el programa generado (esta vez sin la línea 0). Más tarde podrás utilizar tu nuevo programa sin tener el **Beta Basic** cargado.

– Existen muchos más ejemplos de esta segunda forma de utilización. Podemos simplemente usar las facilidades de edición del **Beta Basic** y el movimiento vertical del cursor. O también imagínate que tras la realización de un gran programa necesitas ahorrar memoria. Con el comando **JOIN** podrás unir dos a dos todas las líneas de tu listado y conseguir una macro-línea, ahorrando 5 bytes por cada número de sentencia que elimines. En el campo gráfico ocurre otro tanto. Podemos rellenar figuras de color con **FILL** o alterar atributos con **ALTER** y una vez acabada la pantalla grabarla aparte. Como comprobarás podemos utilizar el **Beta Basic** para conseguir nuestros fines sin tener luego que tenerlo completamente ubicado en la memoria.

Para hacer un resumen de las posibilidades del programa nos centraremos en la versión 1.8 por considerar que es la más difundida en España.

Una vez cargado el **Beta Basic** sabremos que está funcionando al aparecer el mensaje de **Betasoft** en la parte inferior de la pantalla sustituyendo al normal del **Spectrum**. Además el apuntador de línea es parpadeante y se ha aumentado el chasquido de las teclas (que se puede quitar *pokeando* en las variables del sistema). El programa es cargado en la parte alta de la memoria, a partir de la 55800, por lo que en principio no

resulta compatible con interfaces como la RS232 de Indescomp.

Ya estamos en condiciones de poder utilizar los 30 comandos y 20 funciones que añade la versión 1.8 además de otras ventajas que aporta el programa, como pueden ser la aceleración de los programas rodados bajo la acción del **Beta Basic**. Incluso los **GO TO**, **GO SUB** y **RETURN** son más rápidos. Se ha conseguido un **BREAK** más potente que nos puede sacar de algún apuro en código máquina y cuando ejecutemos un **NEW** la línea 0 quedará a salvo.

COMANDOS

Normalmente se accede a ellos pasando al modo **GRAPHICS** y pulsando la tecla correspondiente, y en algunos casos con **GRAPHICS + SYMBOL SHIFT**. Como podrás observar los **UDG** no estarán disponibles a no ser que ejecutemos **KEYWORDS 0**. Los listados estarán más confusos, pero tus programas rodarán perfectamente. Para volver a la normalidad ejecuta el comando **KEYWORDS 1**.

– Facilidades gráficas

* **ALTER (INK, PAPER, FLASH, BRIGHT) TO INK, PAPER, FLASH, BRIGHT** : Permite alterar cualquiera de los atributos en pantalla.

Ej: **ALTER TO PAPER 1, INK 7, BRIGHT 1** cambia toda la pantalla a estos atributos.

ALTER PAPER 2 TO PAPER 1 cambia sólo las posiciones con papel rojo a papel azul.

* **FILL (INK o PAPER); x, y :**

Llena de color el área de una figura que esté en blanco. Las coordenadas x, y son de un punto interior a la figura.

Ej: **FILL x, y** rellena del color de tinta en curso.

FILL INK 4; x, y llena de color verde un recinto.

* **Códigos de control:**

Se han mejorado los pequeños defectos que tenían, como el movimiento a la izquierda en la primera columna (ahora pasa a la fila supe-

rior), o la impresión de "?" en el caso de **CHR\$ 10** y **11**.

CHR\$ 8 cursor izquierda

CHR\$ 9 cursor derecha

CHR\$ 10 cursor abajo

CHR\$ 11 cursor arriba

* **ROLL d (, n)(;x, y; a,1):**

Mueve un píxel la pantalla en la dirección especificada por "d" (las direcciones serán las indicadas por las flechas del **Spectrum**, no como en el caso anterior). Si añadimos "n" la pantalla se moverá "n" píxels. Podemos desplazar sólo una porción de pantalla indicando la esquina superior izquierda (x, y), la anchura ($1 < a < 32$) y la longitud ($1 < l < 176$). Además si sumamos 4 al código de dirección se desplazan también los atributos. Si restamos 4 lo hacen



sólo los atributos. Con ROLL todo lo que se esconde por un lado de la pantalla vuelve a salir por el opuesto.

Ej: ROLL 5, 4 mueve la pantalla 4 pixel a la derecha.

ROLL 6, 8; 0, 175; 32, 88 mueve 8 pixel hacia abajo el cuarto superior izquierdo de la pantalla.

* SCROLL (d)(, n)(;x, y; a, 1):

Casi igual que la anterior. Sólo se diferencia en que ahora con el desplazamiento todo lo que sale por un extremo se pierde y no vuelve a salir. Además podemos escribir tan sólo SCROLL y conseguir que la pantalla se mueva arriba una línea.

* XOS, YOS, XRG, YRG:

Son variables especiales que definen la posición del origen (XOS, YOS) y la escala utilizada en los ejes x

(XRG) e y (YRG). Sus valores normales son XOS=0, YOS=0, XRG=256, YRG=176.

Ej.: LET XOS=55 : LET YOS=82 mueve el origen de (0, 0) a (55, 82) con lo que ahora tendrá sentido PLOT -20, -20.

LET XRG=128: LET YRG=88 aumenta la escala al doble. Si hacemos un DRAW 50,0 con esta escala veremos en pantalla una línea horizontal de 100 pixel de longitud.

* PLOT x, y (; frase) :

Las coordenadas "x" e "y" quedan referidas a la columna superior izquierda. Podemos colocar una frase cualquiera en un pixel determinado, no como con PRINT AT fila, columna.

- Facilidades de edición

* AUTO (x) (, y):

Generación automática de números de línea.

Ej: AUTO empieza en la línea 10 y va de 10 en 10.

AUTO 135 empieza en la línea 135 con un intervalo de 10

AUTO 63,8 empieza en la 63 y sigue con la 71 (de 8 en 8)

* RENUM (x TO y) (LINE z) (STEP s):

Podemos renumerar bloques de "x" a "y" (x TO y). Empezaremos a renumerar en otra línea "z" distinta de la 10 añadiendo (LINE z). El intervalo es introducido con STEP.

Ej: RENUM renumera de 10 en 10 empezando en la línea 10.

RENUM (150 TO 260) STEP 5 renumera el intervalo de 5 en 5 empezando por la línea 150.

RENUM (120 TO) LINE 300 renumera desde la 120 al final, convirtiendo dicha línea en 300, y así sucesivamente.

* DELETE (x) TO (y) :

Borra las líneas de "x" a "y".

Ej: DELETE TO borra todo el programa menos la línea 0.

DELETE 1 TO 1 borra la línea 1.

DELETE TO 230 borra hasta la 230.

* DEF KEY "x"; frase : DEF KEY "x": sentencia

Nos permite introducir en una tecla cualquiera ("x"), una frase o sentencia de frecuente uso. Así cuando pulsemos SYMBOL SHIFT + SPACE + "x" obtendremos la frase o sentencia introducidas. Si lo introducido fue un comando directo, éste se ejecutará.

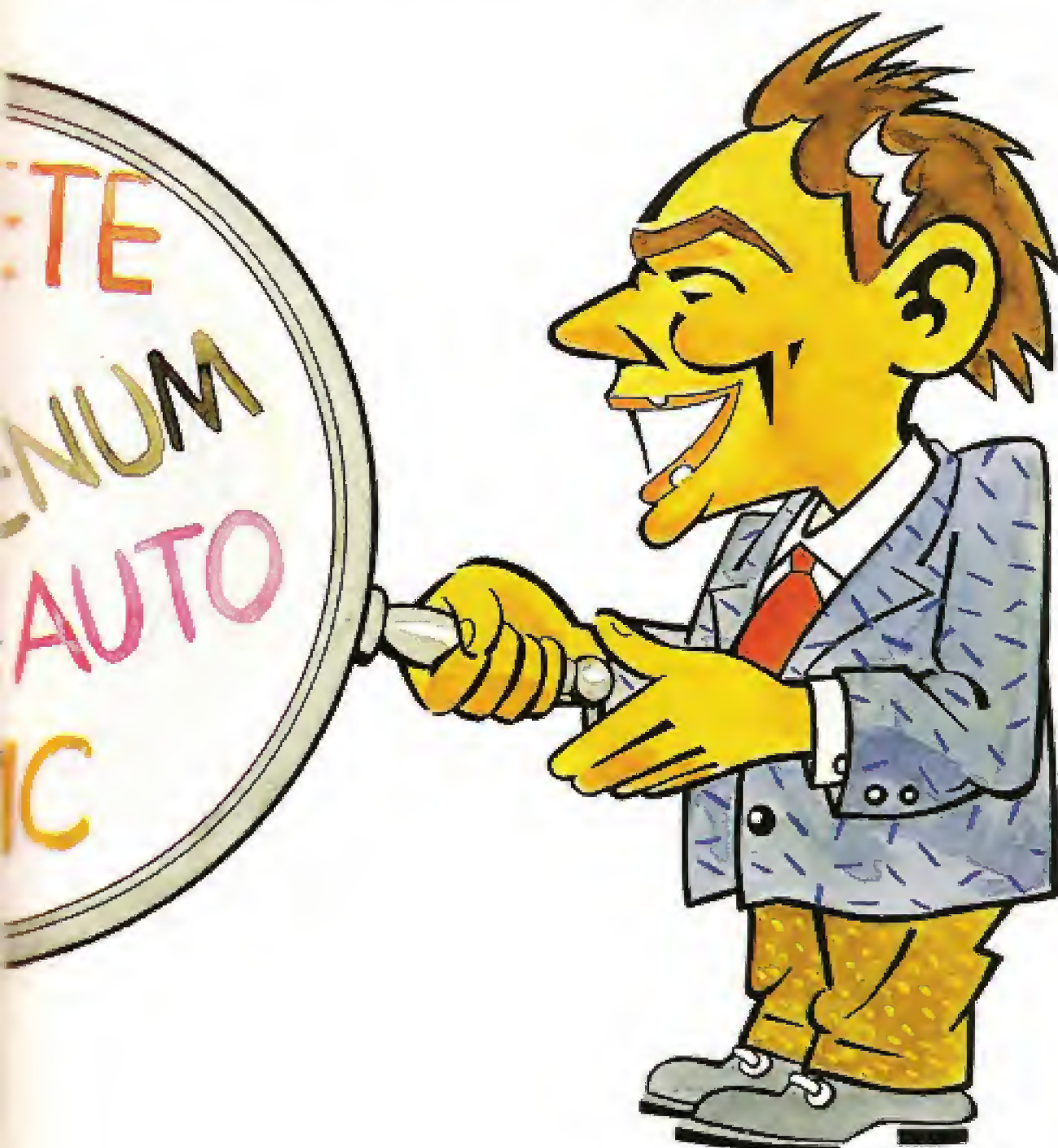
Ej: DEF KEY "8"; "López:" pulsando las teclas S.S.+SPACE+"8" obtendremos en la parte inferior López. DEF KEY "v": PRINT "Pedro" cada vez que lo requiramos imprimirá Pedro. DEF KEY "h"; "10 GO TO n" es similar a las anteriores.

DEF KEY ERASE borra todas las definiciones de tecla.

* EDIT (x) :

"x" es el número de la línea que queremos editar.

Ej: EDIT 50 equivale a LIST 50 + BREAK + CAPS SHIFT 1



* JOIN x :

Une la línea especificada (o si no se hace, la que apunte el cursor) con la inmediatamente inferior, eliminando el número de línea y añadiendo ":".

* SPLIT :

Si en mitad de una multisentencia (después de ":") introducimos el símbolo "<>", conseguiremos dividir la línea original en dos sentencias distintas (con el mismo número).

* LIST x TO y : LLIST x TO y : No necesitan comentarios

* TRACE x :

Es un elemento muy útil para la depuración de programas. Cada vez que se ejecuta una instrucción, si hemos introducido TRACE en el programa se provocará un salto a la subrutina apuntada por "x", lo cual podemos aprovechar para imprimir el número de sentencia, la instrucción ejecutada y las variables utilizadas. Esto es posible ya que son datos que la rutina almacena en las variables LINE y STAT. Podemos conseguir la ralentización de la ejecución de nuestros programas u otros efectos con sólo introducir PAUSE y demás comandos en la subrutina ubicada en "x". TRACE 0 desconecta el comando.

- Facilidades de estructuración

* **Beta Basic** permite una forma de estructuración parecida a GOSUB - RETURN pero mucho más flexible. Se trata de DO - LOOP donde DO simplemente indica el lugar al cual debe regresar la subrutina después de encontrar a LOOP.

Ej: 10 DO

20 PRINT "INPUT": PAUSE 10

30 LOOP

Hasta aquí no es nada del otro mundo. La ventaja reside en la posibilidad de implementar condiciones tanto en DO como en LOOP, y así conseguir una especie de subrutinas autocontroladas.

- DO WHILE (hazlo mientras que...- condición -)

- DO UNTIL (hazlo hasta que...- condición -)

- LOOP WHILE (regresa mientras que...- condición -)

- LOOP UNTIL (regresa hasta que...- condición -)

Ej: Podemos incluir una condición de las dos maneras:

DO UNTIL x < 65 equivale a
DO WHILE x >= 65

Para complementar aún más la estructura podemos utilizar el comando "EXIT IF condición" entre el DO y el LOOP, con lo que conseguiremos abandonar el proceso por otra vía distinta.

* Aparte de la forma de estructuración que acabamos de ver, existen otra serie de comandos que potencian el uso de los bucles en nuestro Spectrum.

- ELSE : Normalmente después del uso de "IF x THEN...", si la condición impuesta no se cumple, todo hipotético comando que se encuentre a continuación separado por ":" dejará de cumplirse. Con ELSE tenemos la oportunidad de ejecutar un comando para ese caso en el que no se cumple el "IF x ..."

Ej: 10 LET k=0

20 IF k<>20 THEN PRINT
"INPUT": ELSE PRINT
"SINCLAIR"

30 LET k=k+1

40 GOTO 20

Para k=20 se imprimirá SINCLAIR en vez de INPUT

- ON: Combinado con GO TO o GOSUB los hace mucho más flexibles. Según el valor que tome la variable situada después de ON, seleccionamos uno de los números de línea de la lista que hayamos introducido. La ventaja reside en que estos números no deben seguir ninguna secuencia lógica.

Como nos indica el manual, es muy útil combinar el uso de ON con GET, que de forma similar a INKEY\$ permite introducir información desde el teclado. La diferencia reside en que GET espera a que se pulse una tecla para continuar (como PAUSE 0). Puede ser usado como variable alfanumérica (GET a\$) o como variable numérica (GET a). Si lo utiliza-

mos de esta segunda forma, cuando pulsemos una tecla numérica obtendremos su valor, pero al apretar A obtendremos 10, con B, 11, etc... (contamos en hexadecimal).

Ej: 10 GET a\$: PRINT a\$;

20 PRINT a\$;

30 GO TO 10

Simulará el efecto de una máquina de escribir.

Ej: 10 GET a

20 GO TO ON a; 123, 41, 2,
123, 43, 54, 63

30 GO TO 10

Cuando pulsemos "2" el GO TO de la línea 20 nos llevará a la sentencia 41. Si pulsamos "6" se producirá un GO TO 54.

* Manejo de errores: ON ERROR x
Comando muy parecido a TRACE en estructura y función. Cuando se produce un error el programa da un salto a la dirección apuntada por "x" y ejecuta la rutina que allí está ubicada. El código del error, la línea y la sentencia donde se ha producido son almacenadas en las variables ERROR, LINE y STAT para su posterior uso. Así, conociendo el código del error a tratar (si consultas el apéndice manual podrás comprobar que el **Beta Basic** añade nuevos errores) se pueden conseguir instrucciones del tipo:

IF error = z AND line = h
THEN RETURN

Si se produce el error de código "z" en la línea "h" el programa regresa a la siguiente instrucción a la que se produjo el error, con lo que se consigue evitarlo.

* Utilizando los procedimientos.

Al igual que en otro conocido lenguaje de programación, el PASCAL, nuestro **Beta Basic** nos permite utilizar los procedimientos (*procedures* en inglés). En esencia se puede considerar que son una especie de subrutinas pero mucho más independientes y de estructura similar a las funciones de nuestro Spectrum. Previamente deberemos definir el procedimiento con "DEF PROC nombre" lo que marca el comienzo de una serie de líneas (no sólo una como con DEF FN) con la

acción a ejecutar, las cuales deberán acabar siempre con el comando **END PROC**. Cada vez que queramos usar el procedimiento lo conseguiremos con "PROC nombre". Las ventajas que aporta esta forma de programación se traducen en una mayor «limpieza» y ordenación de nuestros listados. Con un simple vistazo podemos comprender la estructura del programa, e introducir posibles cambios en un procedimiento sin afectar a los demás, que están perfectamente aislados. La trama ejecutable puede llegar a ser simples llamadas a los procedimientos, los cuales pueden estar definidos en cualquier lugar del listado.

- Usos múltiples

* Los restantes comandos que aporta el **Beta Basic** son una serie de utilidades sobre temas variados:

- **POP x** : Cuando utilizamos **GO SUB**, **DO** o **PROC** se abre una pila donde se anota la dirección a la que debe regresar el control después de encontrar **RETURN**, **LOOP** o **END PROC**. Con "POP x" podemos pasar este valor a la variable "x" para luego poder imprimirlo o manipularlo. Para no alterar la secuencia lógica del programa deberemos añadir "GO TO x+1", ya que **RETURN** no tendría el efecto esperado al no hallar la dirección de retorno en la pila.
- **DPOKE x, n** : Consiste en hacer un doble **POKE**. Nos ahorra el trabajo de tener que dividir un número mayor que 255 en dos bytes (el más y el menos significativo) antes de introducirlo en memoria:

```
10 POKE 30000, 12325-256*(INT 12325/256)
```

```
20 POKE 30001, INT (12325/256)
```

 equivale a:

```
10 DPOKE 30000, 12325
```
- **KEYIN** : Es el ya comentado comando que nos permite la generación automática de sentencias **DATA**, aunque en realidad pueden ser cualquier tipo de líneas, no sólo **DATAs**. En el manual del programa está explicada la rutina generadora de líneas, la cual es de una extraordinaria utilidad.

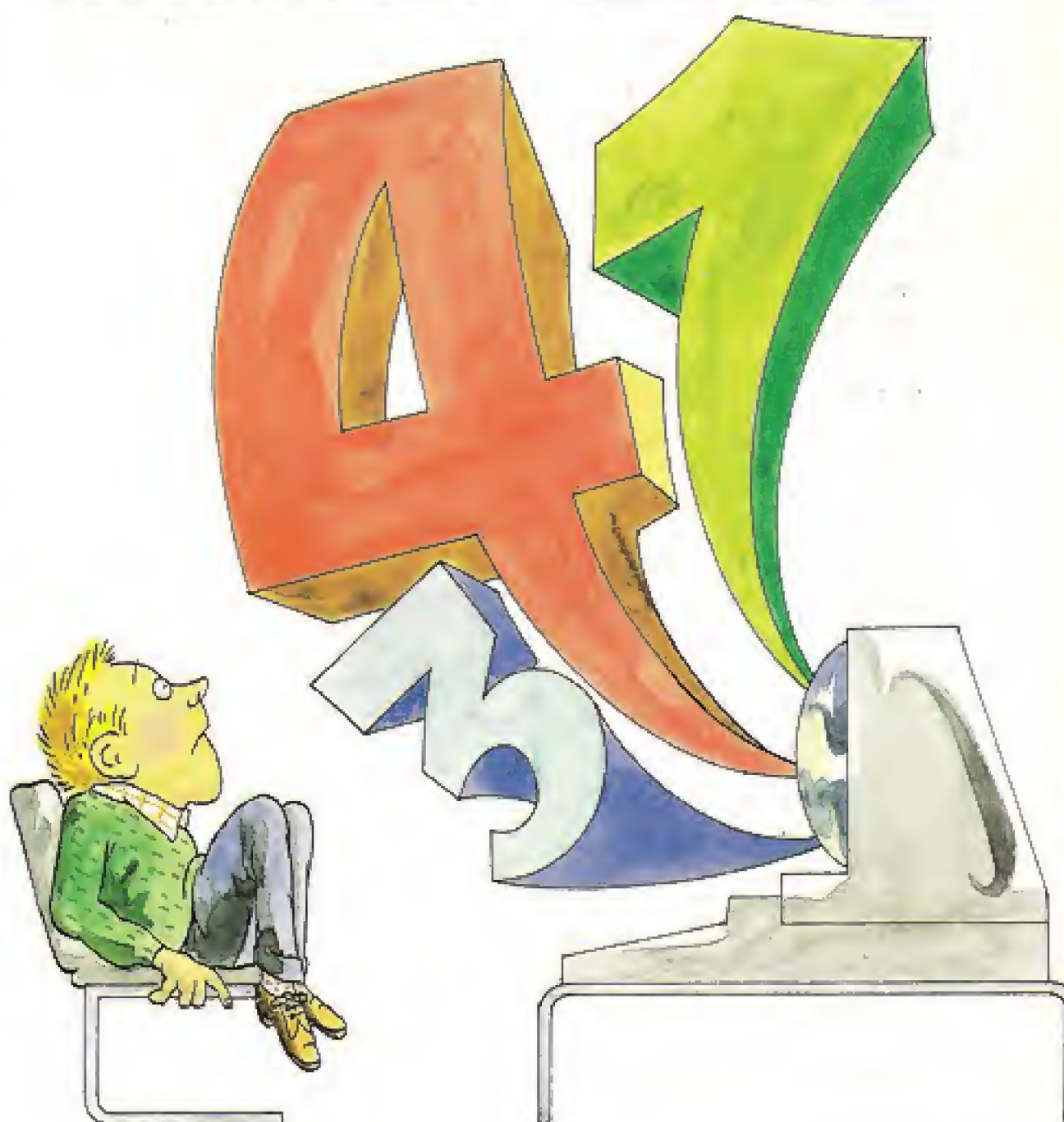
- **CLOCK** : Nos permite la utilización de un reloj de 24 horas que trabaja mediante interrupciones (por lo que sigue avanzando aunque estés ejecutando tus programas). Con sus diferentes opciones podemos conseguir que aparezca el reloj tipo digital en la esquina superior derecha, o suene una alarma a una hora prefijada. La otra posibilidad que incorpora el comando es la de poder ejecutar una subrutina cada vez que transcurra un intervalo de tiempo. La rutina a ser ejecutada no tiene limitación de ningún tipo. Tan sólo hay que tener cuidado cuando ejecutemos **BEEPs** o hagamos cargas de cassette o disco, ya que esto paraliza durante ese tiempo nuestro reloj digital.
- **SORT** o **SORT INVERSE** : Sirve para ordenar en orden ascendente o descendente (**INVERSE**) una

lista de números o letras según su código **ASCII**.

Ej: **INPUT a\$: SORT a\$: PRINT a\$** imprimirá "Mmaaeelnnu" si antes se introdujo "Manuel Mena".

Es un comando muy rápido y potente del que su uso más interesante sea quizá la posibilidad de ordenación de matrices numéricas y alfanuméricas, en su totalidad o sólo una parte (**SORT h\$(1 TO 15)**, **SORT d\$(21 TO)** etc...). Como indica el manual esto podría ser una magnífica ayuda para empezar a crear una pequeña base de datos.

- **USING** : Especifica la forma que deseamos que tengan los números en pantalla. Para definir el formato utilizamos los símbolos "#" (que deja un espacio) o "0" que imprime un 0 siempre que no haya dígitos suficientes para llenar di-



cha posición. Si el formato recorta el número, éste es redondeado, y si una cifra principal (de la izquierda) es truncada, aparece el símbolo "%".

Ej: Un PRINT USING del número 234.306 quedará así con diferentes formatos:

```
"###.##" → "234.31"
"00" → "%34"
"0000.0" → "0234.3"
"&000.000" → "&234.306"
```

– POKE x, string: Permite no sólo la introducción de un dato en la memoria, sino el manejo de una gran cantidad de ellos en combinación con funciones como MEMORY\$(>< x TO y).

Merece la pena nombrar ahora esta función ya que al permitir el manejo de grandes zonas de memoria, va íntimamente unida al uso del citado comando.

El manual desarrolla dos tipos de ejemplo que son dignos de mención. Uno es el manejo de partes de la memoria, como la pantalla o bytes de la ROM que pueden ser movidos de un lugar a otro. Es una simulación desde BASIC del comando en código máquina LDIR.

Ej: Podemos rellenar la pantalla con datos tomados de la ROM consiguiendo una «pintura abstracta»:

POKE 16384, MEMORY\$(>< 1 TO 6912) donde la posición 16384 es la del comienzo del archivo de pantalla a donde van a ser trasladados casi 7K del principio de la memoria.

La otra idea interesante es la de poder preservar programas enteros de la acción de un NEW guardando junto a la zona de memoria ocupada por el programa, las variables del sistema:

CLEAR 33999: POKE 34000, MEMORY\$(>< 23552 TO 33800). Una vez hecho esto el programa estará a salvo de NEW ya que quedará almacenado por encima del RAMTOP. Si más tarde ejecutamos POKE 23552, ME-

MORY\$(>< 34000 TO 44248) lo devolveremos a su lugar.

Al igual que los comandos están realizadas en código máquina, necesitando como ya comentamos la línea 0 para su correcta definición. Se introducen como lo haríamos con una función normal definida por nosotros. Al introducir FN y una de las letras reservadas obtendremos automáticamente la palabra clave. Ej: tecleando FN R< obtenemos RNDM<. El resto de funciones no reservadas se manejan normalmente y funcionan como de costumbre. A continuación adjuntamos un resumen de las funciones clave con un comentario de las operaciones más importantes que realizan.

* AND (x, y), OR (x, y), XOR (x, y): Ejecutan la función lógica que indica su nombre, entre los dos números introducidos.

* SINE, COSE: Funciones seno y coseno modificadas que aunque con un grado de aproximación de sólo 4 dígitos, consiguen una velocidad 6 veces mayor.

* RNDM (x): Da un número aleatorio entre 0 y "x" (es como hacer PRINT INT(x*RND)). Si ejecutamos PRINT RNDM(0) conseguimos un PRINT RND normal, con la salvedad de que es 2 veces y media más rápido.

* DPEEK x: permite un doble PEEK de manera semejante a DPOKE.

* FILLED (): Devuelve el número de pixels rellenados con FILL.

* TIMES(): Almacena la hora que apunta el reloj digital como una cadena de 8 caracteres. Ej: 12:34:45.

* DEC: Conversor hexadecimal-decimal. Ej: PRINT DEC ("1F") = 31.

* HEX\$: Conversor decimal-hexadecimal. Ej: PRINT HEX\$(10) = "0A"

* BIN\$: Conversor decimal-binario. Ej: PRINT BIN\$(4536) nos devolverá el siguiente número en binario: 0001000110111000.

* MEM(): Imprime el número de bytes de memoria que quedan por ocupar. Es equivalente a hacer PRINT 65535-USR 7962.

* MEMORY\$ (): Ya fue comentada junto con el comando POKE.

* MOD (x, y): Devuelve el resto resultante de la operación "x/y".

* CHAR\$ (x): Convierte un número comprendido en el rango 0 a 65535 en dos cadenas de 2 caracteres cada una. Así, el almacenaje en memoria ocupa sólo dos bytes en vez de los 5 que utiliza normalmente, con el consiguiente ahorro de espacio.

Ej: 10 LET H=INT (31255/256):
LET M=31255-H*256

20 LET K\$=CHAR\$ (H)
+CHAR\$ (M)

* NUMBER: Complementa la función anterior. Normalmente la impresión de un número comprimido con CHAR\$ produce un error al contener códigos de control. Con NUMBER podemos devolverlo a su forma normal antes de imprimirlo.

* USING\$: Realiza la misma función que el comando USING pero así como éste sólo podía ser aplicado a PRINT, la función USING\$ permite ser combinada con otro tipo de comandos.

* SCRN\$ (x, y): Muy parecida a SCREEN\$ (x, y) (x=fila, y=columna) pero reconoce también como caracteres normales los UDGs.

* STRING\$: Repite una cadena de caracteres el número de veces especificado, y de forma más rápida que un bucle FOR-NEXT.

Ej: PRINT STRING\$(352, "*")
llena de asteriscos la mitad de la pantalla.

* INSTRING (x, cadena1, cadena2): Busca la serie de caracteres "cadena2" dentro de otra cadena más larga "cadena1" empezando la búsqueda en el carácter "x" de "cadena1". Es posible la utilización del símbolo "#" como comodín.

Ej: PRINT INSTRING (5, A\$, "L#S") provocará la impresión de "LOS", "LAS", "LES", "LDS", etc., siempre que se encuentren a partir del quinto elemento de A\$ (cadena donde buscar).

Las posibilidades de uso son muy amplias pudiendo utilizarse para la búsqueda dentro de matrices, revisión de cadenas introducidas mediante INPUT (y así poderlas comparar), etc...

LOS MEJORES DE INPUT SINCLAIR

PUESTO	TÍTULO	PORCENTAJE
1.º	Commando	23,1 %
2.º	Green Beret	12,2 %
3.º	Saboteur	10,5 %
4.º	Ghost'n Goblins	8,5 %
5.º	Army Moves	7,9 %
6.º	Cobra	7,9 %
7.º	Thanatos	7,7 %
8.º	Rambo	7,5 %
9.º	The Sacred A. of Antiriad	7,4 %
10.º	Batman	7,3 %
		100,0 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «LOS MEJORES INPUT».

Abril de 1987



MAPA, CARGADOR Y POKES PARA...

¡LIVINGSTONE, SUPONGO!

En 1866, el famoso explorador David Livingstone partió del Reino Unido hacia el África negra en busca de las fuentes del Nilo. Tras años sin noticias del explorador, una revista norteamericana encomendó a otro famoso explorador, Henry Morton Stanley, encontrar al supuesto extraviado Livingstone, para lo cual se embarcó hacia África y partió desde Bagamoyo, en la costa Oriental, junto a Zanzíbar, desde donde remontó el río para hallar a Livingstone, dándose la paradoja de que el buscador tuvo que ser buscado. Nuestras aventuras comienzan con las aventuras de Stanley.

Para afrontar nuestra dura empresa, Stanley porta en su mochila un copioso material que nos será muy útil y necesario para llevar a buen fin nuestra misión. Este material se compone de un *boomerang*, con su clásico movimiento de ida y vuelta, un machete, que puede ser arrojado horizontalmente, granadas, las cuales tienen un efecto parabólico, y, por último, una inseparable pértiga, con la cual podre-

mos realizar una infinidad de saltos para salvar los distintos obstáculos.

Todos estos objetos tienen un nivel de fuerza que debemos utilizar perfectamente para conseguir nuestro objetivo. Así que lo primero que haremos es aprender a manejar adecuadamente cada arma.

Acto seguido nuestra misión será la de buscar, pantalla tras pantalla, las cinco gemas sagradas del templo de los Ujiji, pues nos serán imprescindibles para pasar el templo y así poder continuar la aventura. Habremos de estar siempre pendientes de los niveles de nutrición e hidratación, pues de ellos depende nuestra vida. Estos niveles los podremos recuperar recogiendo los diversos alimentos desperdigados por la jungla.

Aparte de los bichejos propios de cada pantalla, y a los cuales poco a poco iremos eliminando, para proseguir nuestro camino hay que hacer especial mención de los siguientes obstáculos. Sin duda, el más temido e incordiante enemigo es el águila. Debemos dejarnos coger sólo una vez y para alcanzar la gema que se halla en

el nido. Una vez recogida, en ningún caso debemos caer de nuevo en sus zarpas, puesto que el águila nos llevaría casi al principio de la jungla, con lo que tendríamos que volver a empezar. Otro obstáculo difícil es el codrilo que se halla en la entrada de la mina y merodea ante ésta. Sólo disponemos de un segundo para introducirnos en la cueva saltando desde el principio de la pantalla. Otro tenaz obstáculo es el Pozo. Debemos calcular muy bien el salto, pues si caemos perderíamos todas nuestras vidas, así que mucho ojo. Por último, las trampas. Deberemos fijarnos mucho en el suelo para no pisar los troncos, puesto que éstos se hunden y caemos en las trampas de las que sólo podremos salir accionando con el *boomerang* las palancas dispuestas a este fin.

Mención especial merecen las pantallas de la cascada, las cuales tendremos que atravesarlas siempre por arriba, pues de lo contrario siempre volveremos al principio. Esto mismo deberemos hacerlo en la última pantalla de la mina.

CARGADOR

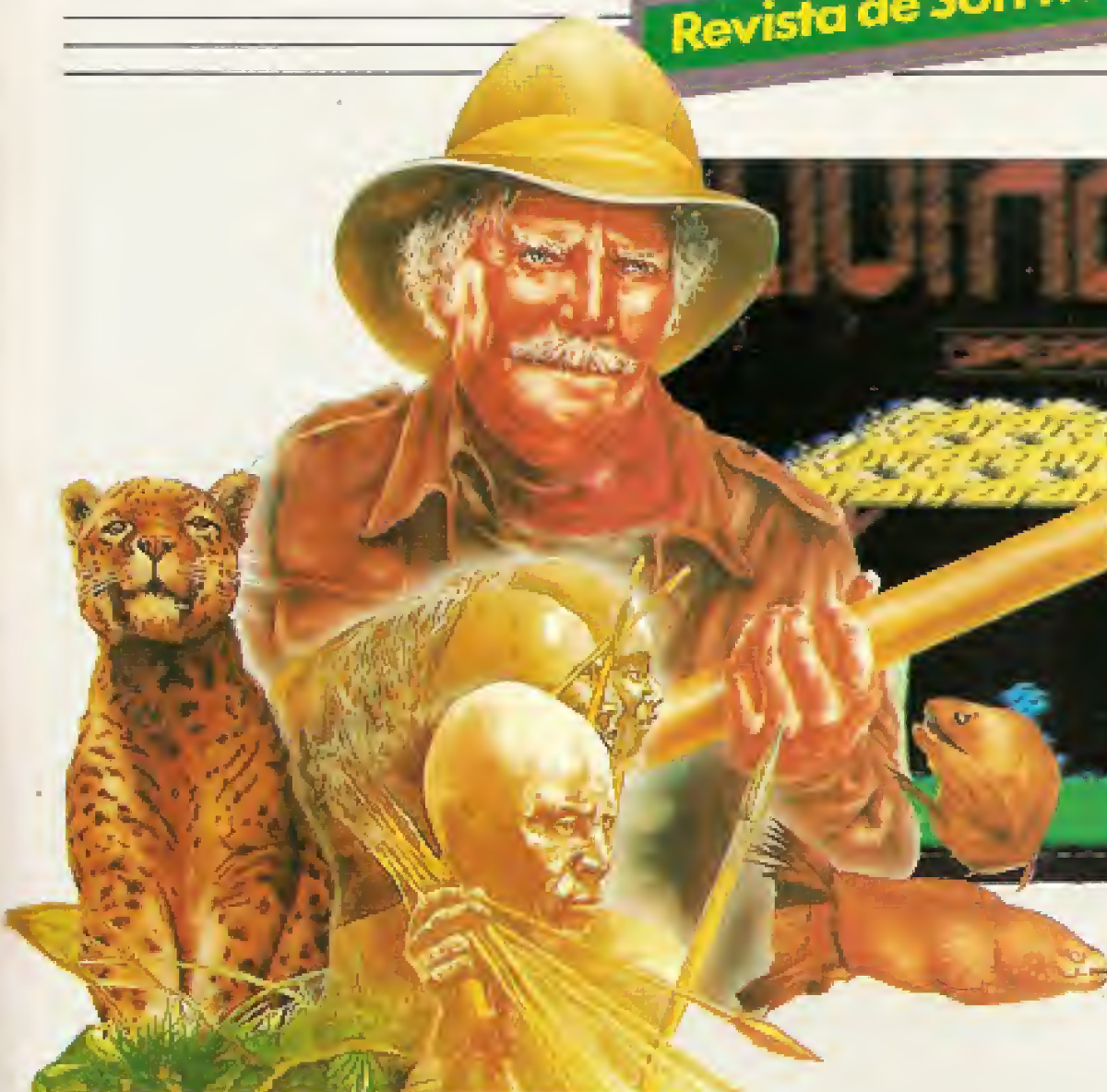
Para hacérselo más fácil, los muchachos de Ópera han incluido en el cargador de su programa una rutina, con la que, al introducir su nombre, «OPERA», a partir de la dirección 62464 nos dará vidas infinitas.

También incluimos un cargador con el que podréis eliminar a vuestro gusto los enemigos.

COMO USARLO

Una vez tecleado, lo salvaremos. Acto seguido lo ejecutamos y el programa nos sacará un menú en el cual elegiremos las opciones que queramos. Finalizadas nuestras peticiones, pulsaremos la «t» e introduciremos el programa ORIGINAL y pulsaremos PLAY; éste se cargará y disfrutaremos del juego a nuestra medida.





```

10 DATA "VIDAS INFINITAS"
20 DATA "AGUILA"
30 DATA "PLANTA CARNIVORA"
40 DATA "OJOS TRAMPA"
50 DATA "MONOS"
60 DATA "MINEROS"
70 DATA "LANZEROS"
80 DATA "ARQUEROS"
90 DATA "MAGO"
100 DATA "VAGONETA"
110 DATA "GRISU"
120 DATA "PRIMITIVOS"
130 DATA "SIRENA"
140 DATA "TORNADO"
150 DATA "BOLA DE FUEGO"
160 DATA "ARAÑA"
170 DATA "MERCENARIO"
180 DATA "COCODRILO"
190 DATA "Pulsar S para
    eliminar TODO"
195 DATA "T-SALIR"
200 BORDER 0: PAPER 0: INK
    7: BRIGHT 1: CLS
205 PRINT " CARGADOR
    LIVINGSTONE SUPONGO ";
    PRINT " -----"
207 PRINT #1;"Pulsar-p-para
    eliminar el bicho": PRINT
    #0;"Pulsar ENTER para

```

```

continuar "
208 LET a=62719
210 READ A$: IF
    A$="T-SALIR"
    THEN RUN
211 LET a=a+1
220 PRINT a$
222 PAUSE 0
223 IF INKEY$="P" THEN GO
    TO 2010
224 IF INKEY$=CHR$ 13 THEN
    GO TO 210
260 IF INKEY$="T" THEN GO
    TO 310
270 IF INKEY$="S" THEN FOR
    m=62720 TO 62720+19:
    POKE m,255: NEXT m: GO
    TO 310
306 IF INKEY$<>"P" OR
    INKEY$<>CHR$ 13 OR
    INKEY$<>"T" OR
    INKEY$<>"S" THEN GO
    TO 222
310 RESTORE 1000: FOR
    J=64000 TO 64095+26:
    READ A: POKE J,A:
    NEXT J
320 CLS : PRINT PAPER 1: INK
    7; #0;" PON EL ORIGINAL Y

```

```

PULSA PLAY ": RANDOMIZE
USR 64000
1000 DATA 221,33,0,0,17,17,
    0,62,0,55,205,86,5
1005 DATA 221,33,0,0,17,
    145,0,62,255,55,205,86,
    5
1007 DATA 243,221,33,0,64,
    17,232,182,62,255,55,
    205,86,5,243,175
1010 DATA 211,254,42,0,245,
    126,183,40,5,42,3,91,
    54,0,221,33
1020 DATA 73,250,253,33,1,
    245,6,17,253,126,0,183,
    40,17,221,126
1030 DATA 0,203,39,95,22,0,
    33,183,213,25,54,159,
    35,54,107,221
1040 DATA 35,253,35,16,227,
    195,0,91,201,7,9,11,12,
    13,14,15
1050 DATA 16,17,24,25,28,
    29,30,33,34,39,44,45,
    49,0,0,0
2010 FOR n=7 TO 0 STEP -1:
    BORDER n: NEXT n: POKE
    a,255: GO TO 210
2020 FOR m=62720 TO
    62720+19: POKE m,255:
    NEXT m: GO TO 310

```


A 2x3 grid of screenshots from the video game Super Mario Bros. The top row shows three different levels: a dark cave with a green pipe, a desert landscape with a green pipe, and a desert landscape with a green pipe. The bottom row shows three different levels: a dark cave with a green pipe, a desert landscape with a green pipe, and a desert landscape with a green pipe.

**M RPA
DEL
LIVINGSTON
SUPONED E**





- A. AGUILA
- B. TRAMPA 1
- C. TRAMPA 2
- D. GEMA
- E. NIDO



FINAL?

LO ULTIMO DE MASTERTRONIC

Hasta ahora, siempre hemos recelado de los programas presentados por firmas que repetidamente atentan contra el buen gusto y la infinita paciencia de nuestros sufridos usuarios. Por ello, cuando una de estas firmas da un giro de ciento ochenta grados, y comienza a ofrecer programas verdaderamente buenos, no nos queda más remedio que aplaudir doblemente el mérito.

Esto es precisamente lo que ha ocurrido con **Mastertronic**, o lo que nuestros amigos de **DRO Soft** han hecho que ocurriera a través de su afortunada intervención. El resultado, comentado telegráficamente, es éste:

180

180 es un interesante simulador de lanzamiento de dardos (si no nos equivocamos, el único del mercado), que incluye varias opciones que te permitirán jugar con los amigos, practicar tranquilamente para perfeccionar tu estilo, o desafiar al ordenador en un campeonato contra ocho contendientes. Con unos gráficos excelentes, y una cuidada presentación, sus autores han sabido resolver brillantemente el problema de reproducir en un ordenador un juego tan peculiar como el de los dardos.



STORM

Un arcade con mucha acción y un tema casi nostálgico: rescatar a la princesa cautiva.

Si lo que quieres es pasar un buen rato sin complicarte la vida, o simplemente estás harto de complejas vídeo-aventuras de las que no se resuelven ni a base de pokes, entonces seguro que te gustará **STORM**.

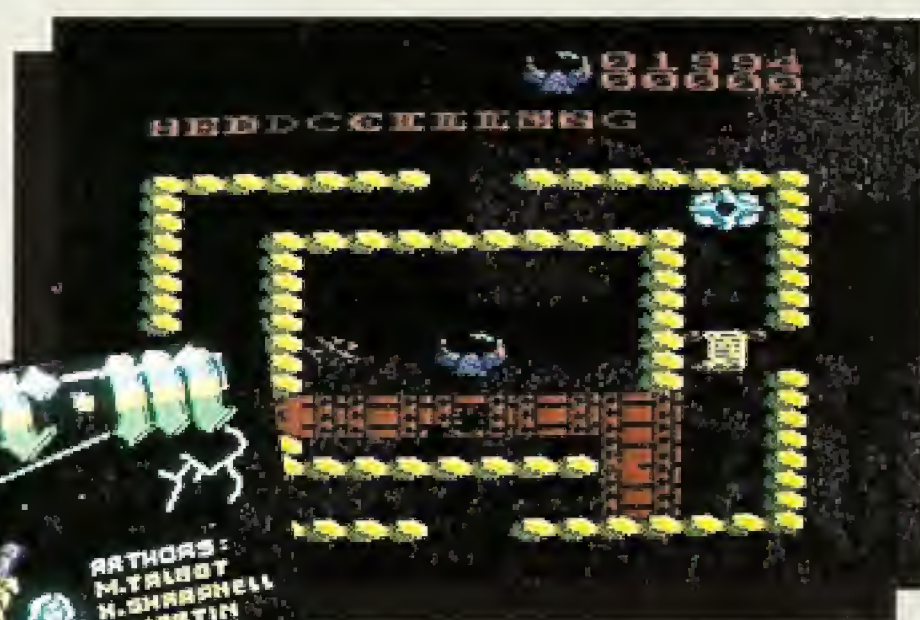


BUMP, SET, SPIKE!!

Este programa reproduce una modalidad deportiva casi

desconocida en Europa, que en los últimos años ha alcanzado grandes cotas de popularidad en los EE. UU. Se trata del *Volleyball-dobles*.

El planteamiento gráfico del juego es similar al empleado hasta ahora para otros deportes de

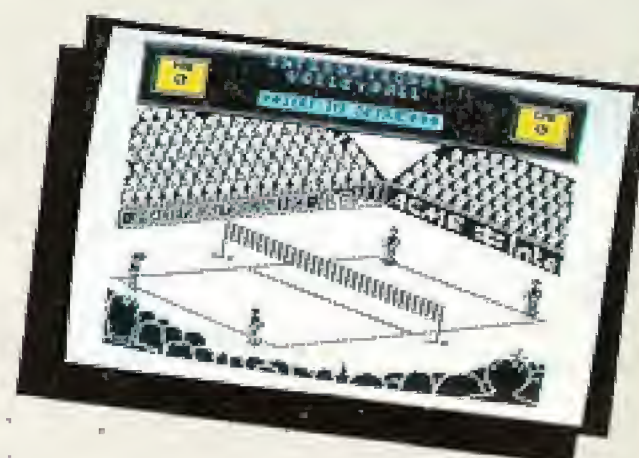


equipo, aunque esta vez se ha aplicado un nuevo punto de perspectiva. La animación de los jugadores y la reproducción de las distintas tácticas de juego, son los aspectos en los que más destaca este original simulador deportivo.

UN SUPER-WARGAME DE 128 K

Recientemente, la compañía inglesa CCS ha lanzado al mercado un sorprendente simulador bélico codificado especialmente para el Spectrum 128 K (y por tanto, incompatible con el modelo de 48), en el que se revive la sangrienta batalla de Gallipoli, una de las acciones estratégicas más interesantes de la Primera Guerra Mundial, librada en 1915 en la orilla europea de los Dardanelos entre las tropas francobritánicas y turcas.

Poco a poco, las compañías productoras están desterrando la idea de que sólo los programas de 48 K son rentables. El hecho que en mayor medida ha contribuido a este cambio de comportamiento, ha sido el incremento de las ventas del Spectrum Plus-2.



IMPOSSABALL

● HEWSON ■ REFLEJOS



Varios números atrás, hicimos un breve comentario de BOUNDER, un simpático programa que nos dejó sorprendidos por su tremenda originalidad e innovadora realización.

Su protagonista, como seguramente recordarás muy bien, era una pelota de tenis extraviada, que se desplazaba botando de un lado para

otro a través de un camino sembrado de obstáculos, tratando de llegar lo más lejos posible.

Desde que se publicó BOUNDER hasta ahora, se han ensayado numerosas interpretaciones a partir de este original modelo, algunas de ellas muy interesantes y con un nivel de calidad nada desdeñable, pero creemos que ninguna ha tenido tanto acierto ni tanta personalidad propia como IMPOSSABALL, un programa presentado por la prestigiosa firma Hewson a finales del año pasado, y publicado en España en fechas muy recientes.

Con unos gráficos de gran calidad y una sorprendente precisión en los desplazamientos, IMPOSSABALL pondrá a prueba tus reflejos y tu



HEWSON
PRESENTS
IMPOSSABALL

BY

JMP 1986

NOW LOADING ■■■■

capacidad para calcular distancias y trayectorias.

Te lo recomendamos en la seguridad absoluta de que este programa creado por Hewson te hará pasar momentos realmente apasionantes delante de la pantalla de tu micro.

ANIMACION	8
INTERES	7
GRAFICOS	8
COLOR	6
SONIDO	6
TOTAL	35



DAN DARE, PILOTO DEL FUTURO

● VIRGIN GAMES ■ CIENCIA FICCION

El 8 de julio de 1985, falleció Frank Hampston, creador de las famosas tiras de **Dan Dare** (desgraciadamente no tan famosas en España), y padre, junto con los autores de Flash Gordon, de la historieta S.F. clásica. Un año después de su muerte, la firma Virgin Games adquirió los derechos para reproducir su última aventura a modo de homenaje póstumo, respetando en la medida de lo posible los dibujos y el planteamiento de la acción característicos del genial estilo de Frank Hampston. El resultado, en sus tres versiones (Spectrum, Amstrad y Commodore) no ha podido ser mejor, sobre todo si tenemos en cuenta que en España podemos disfrutar de una versión en

castellano, traducida y adaptada por DRO Soft.

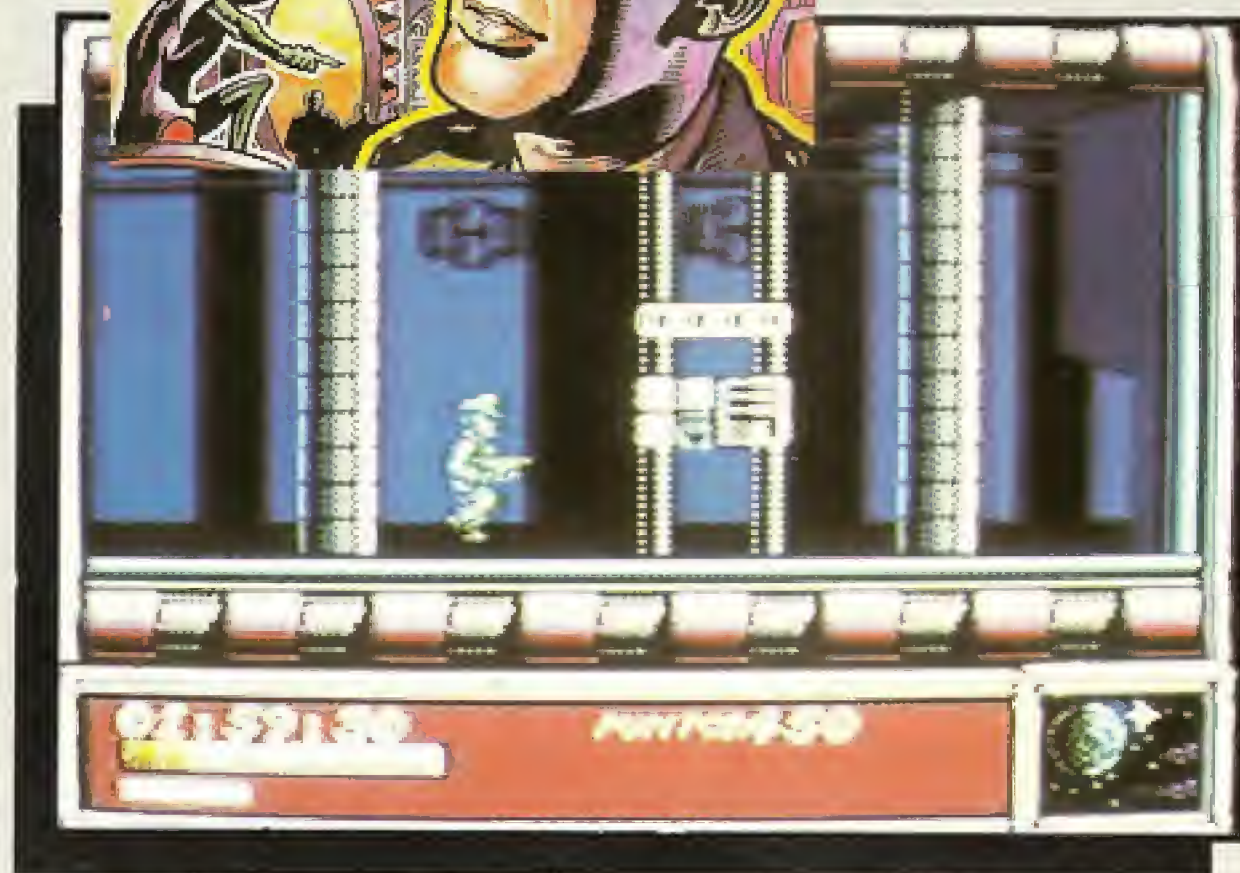
El argumento, propio del más puro género, se desarrolla dentro de los cánones del cómic clásico de aventuras, sin perder por ello ciertos rasgos de originalidad. El planteamiento inicial es el siguiente: El malvado **Mekon**, aspirante de turno a «Señor del Universo», ha dado un cruel ultimátum a la Tierra:

si los gobernantes del mundo no acceden a rendirse sin condiciones, el planeta será destruido por un asteroide que se aproxima a gran velocidad. La única forma de evitar la fatal

colisión, es hacerlo explotar antes de que sea demasiado tarde, pero sólo **Mekon** conoce los lugares donde se esconden los componentes de la secuencia de autodestrucción.

La misión de **Dan** consiste en llegar hasta el asteroide para encontrar dichos componentes, e insertarlos en el panel de control adecuado. En su camino, tendrá que derrotar a los esbirros que intenten capturarlo o restarle energía, sortear peligrosos obstáculos, y combatir contra el propio **Mekon**, que tratará por todos los medios de evitar que sus ambiciosos planes fracasen.

DAN DARE Piloto del Futuro es, sin duda, una de las mejores vídeo-aventuras de la temporada. La calidad excepcional de sus *sprites*, la originalidad y el acierto del diseño gráfico, y su elevado grado de interés, la hacen acreedora de una valoración extraordinariamente positiva. Te la recomendamos sin reserva alguna.



ANIMACION	8
INTERES	9
GRAFICOS	9
COLOR	9
SONIDO	7
TOTAL	42

ACROJET

● MICRO PROSE ■ ACROBACIA AEREA

«El BDS-J ACROJET es un reactor monoplaa especialmente diseñado para la acrobacia aérea, capaz de alcanzar velocidades superiores a las 200 mph. Su gran maniobrabilidad y su sofisticada instrumentación, le convierten en una de las máquinas más perfectas construidas por el hombre.»

La firma MICRO PROSE, especializada en programas de simulación, acaba de presentar en



comentario de siempre: «parece mentira lo que se puede llegar a hacer con 48 K escasos».

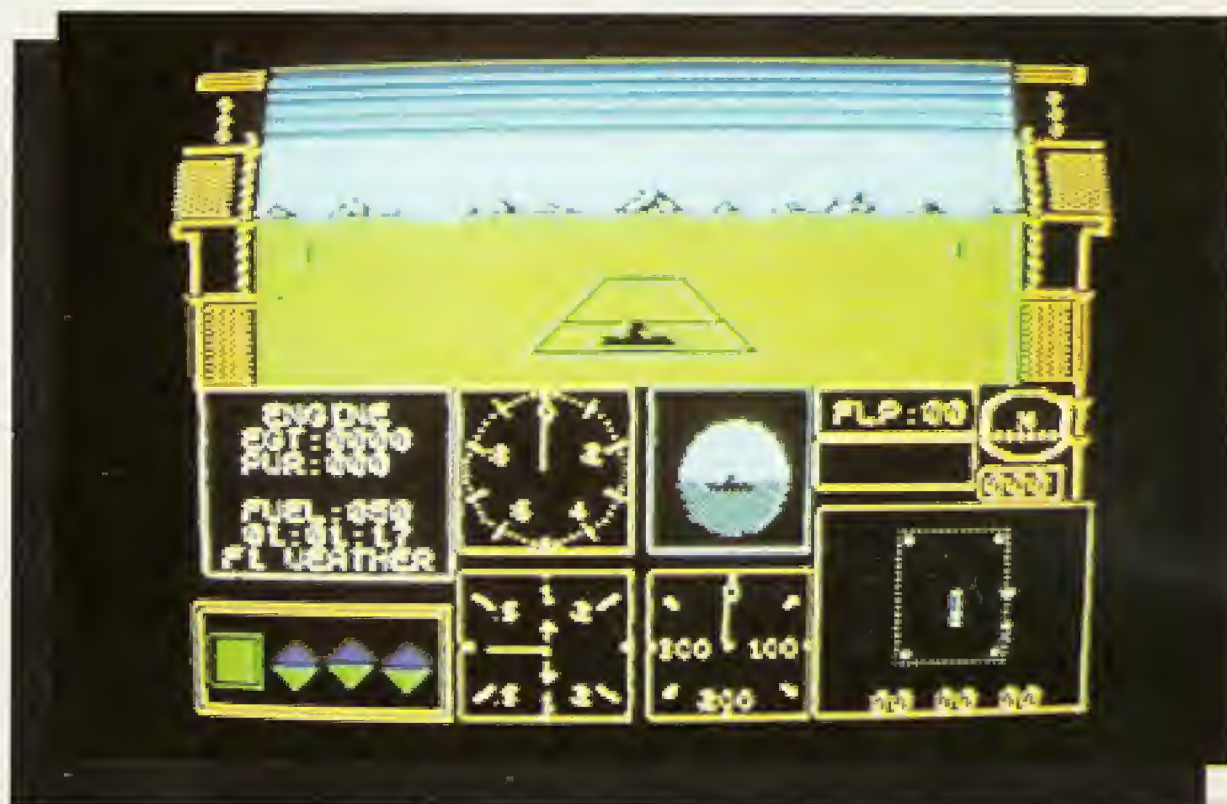
Seguramente, pronto vendrá otro simulador aún mejor dispuesto a jubilar a éste, pero mientras tanto, podremos asegurar que ACROJET es el más completo de cuantos se han hecho.

No te lo pierdas.

ANIMACION	8
INTERES	9
GRAFICOS	9
COLOR	8
SONIDO	6
TOTAL	40

España un sofisticado simulador de vuelo acrobático, con el que se ha logrado alcanzar un nivel de realismo hasta ahora impensable en un microordenador.

Cuando se publicaron **Strike Force Harrier** y **Ace**, creímos que ya se había dicho la última palabra sobre este tipo de programas. Recuerda que al primero de estos programas lo calificamos nada menos que como «sensacional» y que le enviamos nuestras más efusivas felicitaciones a la firma fabricante, Mirrorsoft. Pero, pero como suele ocurrir en estos casos, el tiempo ha venido a demostrarnos lo contrario, y de nuevo hemos de repetir el



XEVIOUS

● US. GOLD ■ ARCADE

Se trata de un arcade del más viejo estilo, respetuoso con los cánones clásicos, pero falto de *gancho* adictivo. Los gráficos y la animación están en un nivel aceptable, pero el conjunto no llega a ser lo suficientemente bueno como para merecer una valoración tan favorable como se esperaba.

El argumento, como corresponde en

ANIMACION	5
INTERES	7
GRAFICOS	7
COLOR	6
SONIDO	6
TOTAL	31



estos casos, es bastante simple. La misión consiste en tripular una sofisticada super-nave para acabar con los Xevious, unos perversos seres que codician las riquezas de la Tierra, y que se disponen a invadir nuestro planeta.



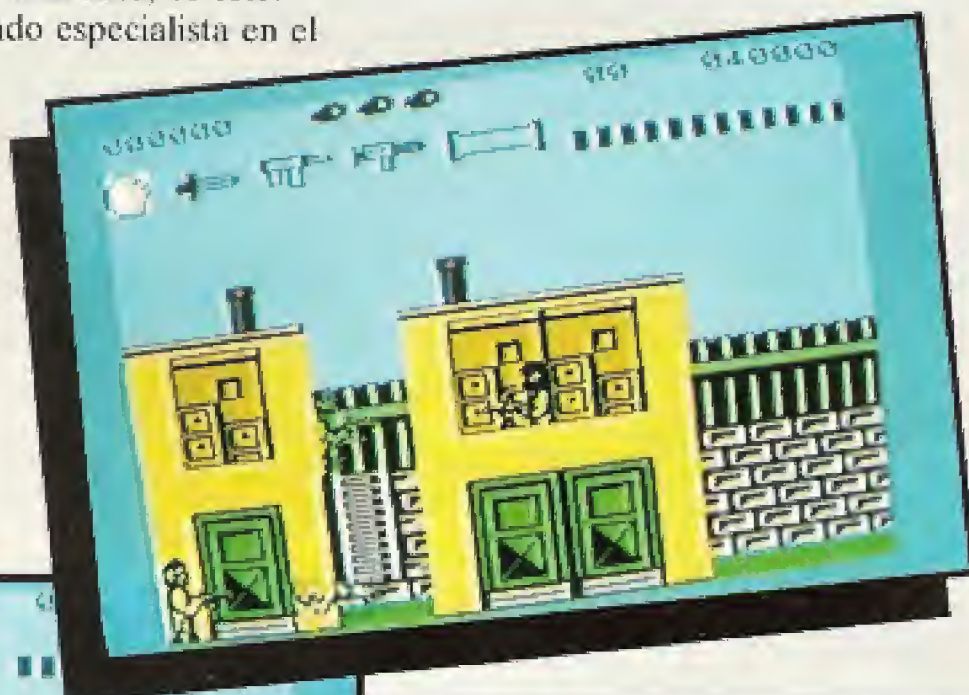
BAZOOKA BILL

● ARCADE SOFT ■ ARCADE

A pesar de su escasísima originalidad, BAZOOKA BILL reúne todos los ingredientes necesarios

para crear una persistente adicción, especialmente en aquellos que saben disfrutar a fondo un arcade sin preocuparse mucho por su calidad. El argumento, extraordinariamente parecido al de otros programas de características similares, es éste: Un supersoldado especialista en el

manejo del bazooka ha sido elegido para una difícil misión. Armado hasta los dientes, se dispone a participar en el más sangriento combate cuerpo a cuerpo, y a pilotar un sofisticado caza de combate.



ANIMACION	4
INTERES	7
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	31

DEACTIVATORS

● ARIOLA SOFT ■ DESACTIVACION DE BOMBAS

«Alerta!! Un comando terrorista se ha infiltrado en el Instituto de Investigaciones Gravitacionales para sabotear sus instalaciones. Se han detectado un gran número de dispositivos de detonación colocados en diversos lugares del edificio, preparados para explotar en cualquier momento.»

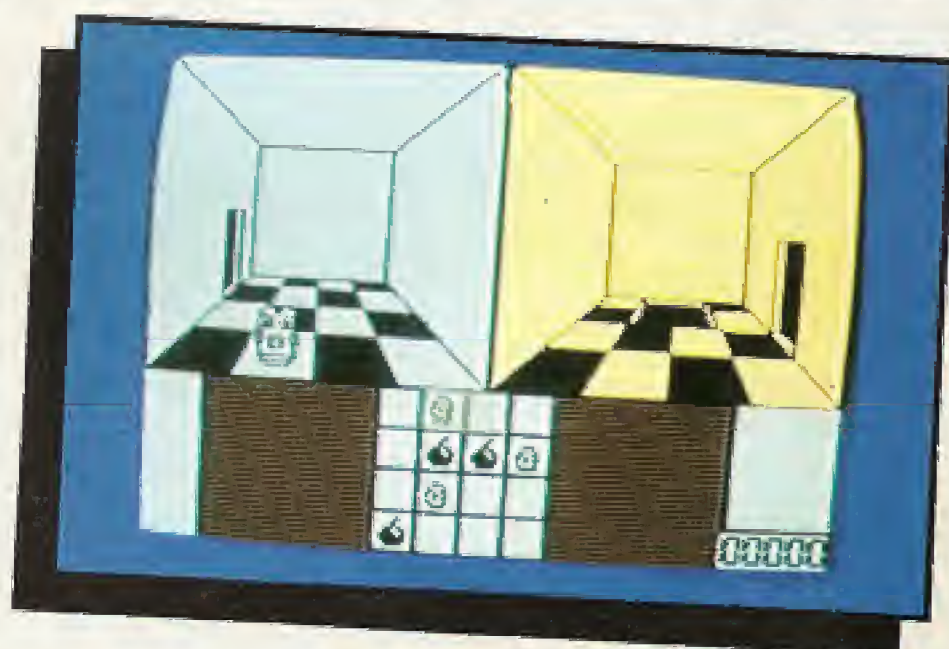
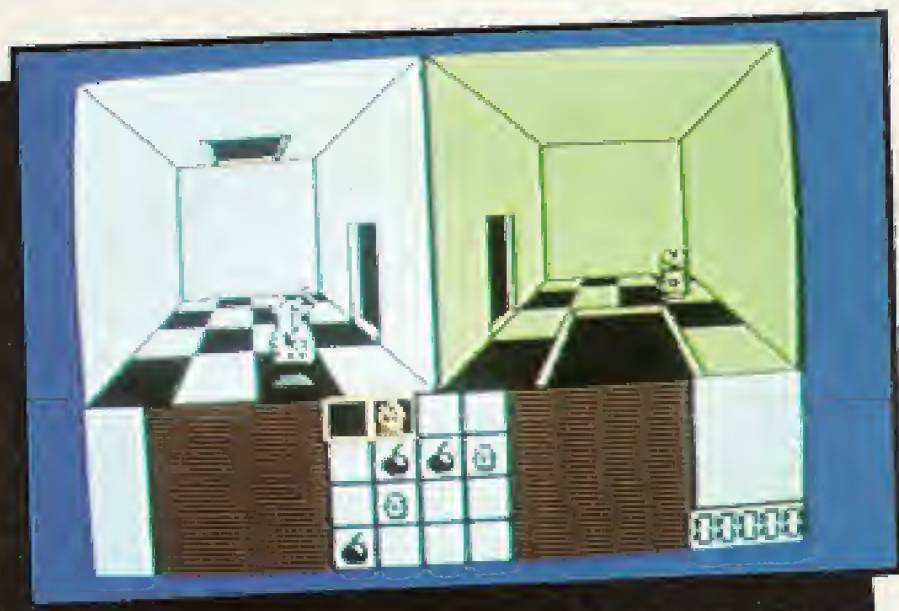
Cuando el jefe de seguridad leyó el mensaje emitido por el sistema de detección y alarma, supo que por fin tenía en sus manos una misión a la

medida de los Deactivators, un equipo de androides especializados en trabajos de desactivación. Sin pensárselo dos veces, conectó sus circuitos y los programó para realizar la difícil tarea...

El objeto del juego consiste en sacar todas las bombas de los cinco niveles que componen el edificio, arrojándolas fuera antes del tiempo límite. Además, habrá que reparar los elementos disfuncionales que han alterado el equilibrio gravitacional de las habitaciones, y evitar el contacto con los robots reprogramados por los terroristas. **Deactivators** es un programa difícil de comprender al principio —quizá

por la originalidad de sus planteamientos— y en cierto modo falta de esos pequeños detalles «invisibles» que retienen la atención del jugador. No obstante, basta con echar un vistazo a sus pantallas para comprobar que calidad gráfica no falta. En definitiva, un programa recomendable, con ciertas reservas.

ANIMACION	9
INTERES	7
GRAFICOS	8
COLOR	6
SONIDO	6
TOTAL	36



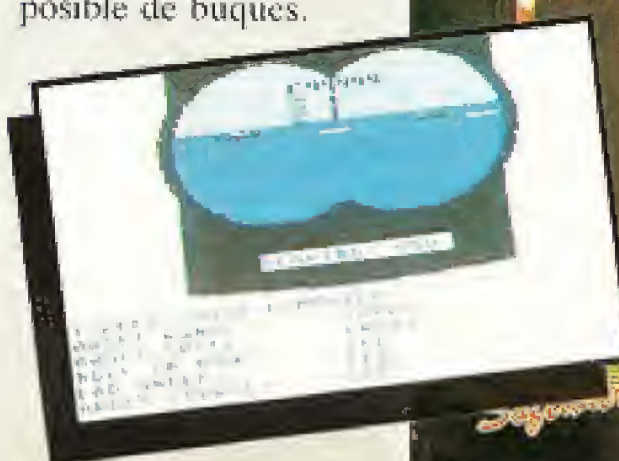
SILENT SERVICE

● MICRO PROSE ■ COMBATE NAVAL

Silent Service es un excelente programa de ambientación bélica, en el que se revive toda la acción y la estrategia que tuvo lugar en las aguas del Pacífico durante la Segunda Guerra Mundial. Desde la sala de máquinas, la torre de control o el puente de mando,

podrás poner a prueba tu habilidad como comandante de uno de los submarinos que patrullaban las rutas de navegación de la escuadra nipona. Tu misión: aproximarte a la flota enemiga esquivando las barreras de cargas de profundidad, y hundir el mayor número posible de buques.

ANIMACION	8
INTERES	8
GRAFICOS	8
COLOR	7
SONIDO	6
TOTAL	37



SILENT SERVICE



FIST II

● MELBOURNE HOUSE ■ ARTES MARCIALES

Seguro que muchos de vosotros conserváis un grato recuerdo de aquel sensacional programa que fue «The way of the exploding fist», un número uno en ventas que pronto acabó cayendo en el olvido. Pues bien, su esperada segunda parte ya es una realidad.

ANIMACION	8
INTERES	7
GRAFICOS	8
COLOR	6
SONIDO	6
TOTAL	35

Bajo el título —más breve esta vez— de FIST II, Melbourne House nos ofrece mucho más que una mera continuación de la aventura. Se ha hecho una reinterpretación total del desarrollo original del juego, añadiendo un argumento, un escenario-laberinto móvil (nada de pantallas de fondo fijo), y una mayor diversidad de

oponentes. Con gran acierto, sólo se ha conservado de la versión precedente lo mejor: la calidad y profusión gráfica de los golpes y movimientos en el combate. Por si esto fuera poco, se ha incrementado aún más la velocidad de los desplazamientos y la resistencia de los contendientes. En definitiva, FIST II es una versión en clave de aventura de lo que inicialmente fue un simple simulador de artes marciales.



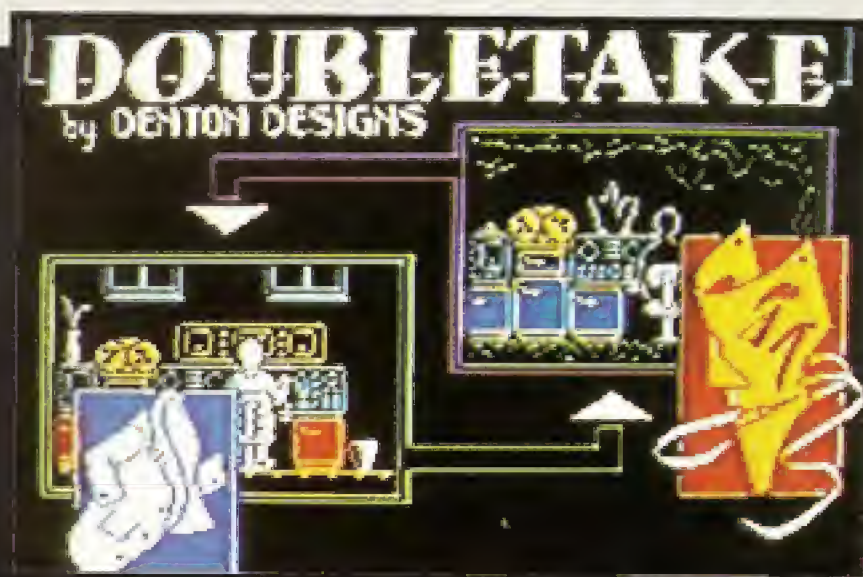
DOUBLE TAKE

● OCEAN ■ AVENTURA FANTASTICA

La proverbial originalidad de la firma británica Ocean Software ha dado recientemente nuevos frutos. En esta ocasión, se trata de un excelente programa en el que se aúnan unos buenos gráficos y un argumento original como pocos.

En el año 2008, se abre la única puerta que comunica la Tierra con un universo negativo paralelo al nuestro. Cada objeto de este mundo tiene en el otro una réplica exacta en negativo, pero ahora, después de la colisión, ambos universos han perdido estabilidad,

y los objetos han comenzado a pasar de un lado a otro, mezclándose en un caos cósmico del que sólo tú puedes librarnos. Original. ¿No?



ANIMACION	9
INTERES	8
GRAFICOS	9
COLOR	8
SONIDO	7
TOTAL	41



EL ZOCO



VENDO ZX Spectrum Plus, con todos sus accesorios y manuales, más interface Kempston y Joystick Quick Shot 11, todo en perfecto estado y con embalajes originales; revistas Microhobby (colección completa) con curso Basic y C/M, juegos y curso de Basic de Electrónica. Precio estupendo. Llamar al tel. (976) 51 07 15. Zaragoza.

COMPRO para el ZX Spectrum programas de utilidades y gestión en especial compiladores para «C» y «COBOL». También una impresora matricial y unidad de discos. Manuel Blanco Vidal, c/. Real n.º 4 Puebla de Trives. 32780 Orense.

VENDO Interface de Joystick programable por 3 000 pts. Jorge Díaz García. Avda. de Santander, n.º 12, 6.º A. Tel. 22 35 89. 33001 Oviedo.

VENDO ordenador Spectrum 48K, cassette, más de 200 juegos y aplicaciones libros y revistas por 35 000 pts. Llamar al tel. 253 05 29. Javi.

VENDO ordenador ZX Spectrum, con todos sus cables, cinta de demostración y libro de instrucciones. Poco uso. Todo por 20 000 pts. negociables. Llamar al tel. (96) 365 54 25. Daniel.

VENDO ZX Spectrum 48K en muy buen estado. Manual, cables de conexión, Joystick, siete libros y más de veinte revistas (incluidos cinco números de INPUT SINCLAIR). Regalo además ZX 81 con ampliación a 16K (manual, cables, etc...). Todo por 19 000 pts. Llamar al tel. (93) 217 07 15. Juan.

VENTA E INTERCAMBIO programas 16/48K. Tengo últimas novedades. Escribir a: Rubén Herrero Angulo, c/ Arenal 105, 1.º, n.º 1. 09200 Miranda de Ebro. Burgos. Tel. (947) 31 35 61.

VENDO interface Joystick programable tipo Kempston por 3 000 pts.; comprado en septiembre. Escribir a: Carlos Bernues Ayerza, c/ General Franco, n.º 43. 22600 Sabiñánigo. (Huesca).

CAMBIO programas. Escribir a: Francisco Muñoz García, P.º Zorrilla, n.º 5, 1.º B. 47007 Valladolid.

INTERCAMBIO juegos, poseo 400 la mayoría novedades. Me interesan (G-Joe, The Evil Crown, Zakii Wood, Black Wiche, Back to School, Alien, Temple of Apshai Trilogy, Fist 11, Conflict in Vietnam y Piracy). Escribeme mandando lista. Pablo Castaño, c/ Maestro Guerrero, n.º 8, 7.º 4.º. 29002 Málaga.

VENDO ordenador ZX Spectrum 48K completo, nuevo, con 15 cintas de programas, libros, cassettes c-15 vírgenes, porta cassette, etc. Por sólo 15 000 pts. Enciclopedia RUN, 4 tomos encuadrada a convenir. José Manuel Pérez.

Aptdo. 42, Lugones. Asturias. Tel. (985) 26 07 94.

VENDO Spectrum Plus, nuevo, Interface Kempston, Joystick, 10 programas originales (Skyfox, Match Point, etc.) 20 o más revistas varias, 3 libros, 25 cintas (280 juegos). Todo ello por 42 000 pts. negociables. Francisco Cosano Cabanillas. Pje. Pintor Córdoba, n.º 1, 1.º 1.º. 14006 Córdoba. Tel. (957) 27 32 41.

VENDO ordenador LASER color computer 200, completamente nuevo y sin usar; con adaptador, cables, cassette de demostración, manuales, etc. Precio 20 000 pts. Negociables. Emma Carballal Haire. La Caeyra, Pza. Riestra, n.º 2, 12.º 36005 Pontevedra. Tel. (986) 85 68 80.

VENDO ZX Spectrum Plus y un cassette para ordenador, todo en buen estado, por 29 000 pts. Regalo Interface para Joystick y 10 juegos a elegir. Escribir a: Fernando Sánchez Palomo, c/ Marqués de la Valdavia, n.º 17, 2.º-B. 28100 Alcobendas. Madrid.

VENDO/CAMBIO juegos. Tengo los mejores (Asterix, Jack the Nipper, Pentagon, Heavy to the Magic, Equinox, entre otros). Alfredo Ruiz Gutiérrez. Avda. Madrid, n.º 15, 4.º-D. Granada. Tel. (958) 27 23 63.

CLUB de intercambio, venta compra de SOFTWARE. Francisco Puig Alarcón, c/ Paseo Estación, n.º 37, 3.º 2.º. 25600 Balaguer. Lérida.

Pensionista 53 años, gustos electrónica y micros, agradeceré: regaléis revistas, material, piezas, aparatos, etc. Remitir a: Julián Seguen. Serradilla, n.º 28. 28044 Madrid. Tel. 705 98 24.

GANADORES DE LOS MEJORES DE INPUT SINCLAIR

En el sorteo correspondiente al número 18 entre quienes escribisteis mandando vuestros votos a **LOS MEJORES DE INPUT** han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
ESTEBAN CIVERA SALTO	Gavá (Barcelona)	WINTER GAMES
JUAN JOSÉ COLOMINA VERT	Porto Cristo (Mallorca)	PIRACUSE
ARTURO ROBÍO PAVÓN	Burgos	IKARI
JOSÉ I. GUDE BASTERRECHEA	Madrid	INFILTRATOR
JOSÉ LÓPEZ FONTANET	Algemesi (Valencia)	FAIR LIGHT II
LUIS ESTEBAN MANZANARES	Navahermosa (Toledo)	DEEP STRIKE
JOSÉ M.ª MARTÍNEZ MORENO	Puerto Sta. María (Cádiz)	YIE AR KUNG-FU
JOSÉ I. SÁEZ MAESTRE	Madrid	SHADOW OF THE UNICORN
LUIS FELIPE GARCÍA ARCAÑA	Benicasin (Castellón)	THANATOS
JUAN J. SAURA PAREDES	Elche (Alicante)	STRIKE FORCE HARRIER

EL ZOCO

Vendo ordenador Apple II E 64 K Spanish, Disco con Controlador, Disco sin controlador, Tarjeta de Ordenador a Impresora, Joystick Apple II E, con quince programas importantes, TODO SIN ESTRENAR, con factura; hago 50 % descuento. Contactar con Antonio Benítez. Avda. Dr. Solís Pascual, 28. 11600 Ubrique (Cádiz). Tl.: 956-11 01 59.

Doy clases de Basic, Ensamblador, Cobol, Pascal y Fortran 1v, además de Programación Estructurada en Basic y Matemáticas de Computación. Sólo para Madrid. Llamar al tl.: 201 66 02. Ángel Durán García.

Vendo Spectrum 128K por 30.000 ptas., con todos sus Accesorios (transformador, cables y manuales) y 20 juegos (Antirad, Uridium, Thanatos, etc.). Escribir a: Manuel Ríaza Huget. C/ San Antonio M.^a Claret, 39. Barcelona 08025. Tl.: 93-207 03 01 de 3 a 8.

Vendo ZX Spectrum Plus, manuales en castellano, lápiz óptico y cintas 18.000 ptas. Llamar al tl.: 695 78 17. Getafe. Madrid. Marcos Cabañas.

Vendo el libro de «Código Máquina del Spectrum» P.V. 1.500 ptas. y regalo «Código máquina para principiantes». «Sprites y gráficos en lenguaje máquina» P.V. 1.500 ptas. y regalo n.º 23, 24 todo Spectrum. Vendo números 3, 5, 8, especial verano, 10, 11, 13, 14 de Input Sinclair cada uno por 275 ptas. Llamar al 85 52 40. J. Javier Casado. Cádiz.

Últimas novedades en Software y por cariño interesado en conseguir el programa «World Cup Carnival». Francisco Crusat. C/ Serafin Avendaño, 2, 3.º A. Vigo 36201. Pontevedra. Tl.: 986-22 25 73. Lunes y jueves de 7 a 9.

Desearía contactar con usuarios del Spectrum, para intercambiar, comprar o vender programas, trucos, libros e información. Antonio Sáez Bravo. C/ Alberche, 136. 45007 Toledo. Tl.: 925-23 15 62.

Cambio, compro y vendo programas para Spectrum 48K, primeros juegos. Daniel Luna. C/ San Antonio, 37, 3.º Vilaseca, Tarragona. Tl.: 977-39 03 08.

Intercambio toda clase de programas, mapas, trucos, ideas y pokes. Alex Bernal Grau. C/ Idumea, 6-8, 5.º-6.º 08035 Barcelona. Tl.: 93-212 55 44.

Vendo juegos por cambio de ordenador. Top Gun, Pyracuse, Commando, Green Beret, Winter Games, Bom Jack entre otros. A 600 ptas. c/u. Mando

lista; gastos de contrarreembolso a mi cargo. Enviar tl. de contacto. Gregorio Riquelme. P.º Alfonso XII, 54, 5.º B. Cartagena 30203.

Compro cassette Gold King. Llamar tardes de 7 a 8. Tl.: 433 52 95. Adolfo. Madrid.

Cambio programas de Spectrum 48K. Enviar lista. Tengo novedades: Trivial Pursuit, Uridium, Thanatos, etc. Pierre Arcelin Zabala. C/ Castillo Palautordera, 86. Vilafortuny (Cambrils). Tarragona. Tl.: 977-36 13 02.

Vendo ZX Spectrum 48K Plus; con el teclado español, manual, cables de conexión y dos cintas, una con 10 juegos; y la guía de funcionamiento. Todo por 22.000 ptas. Carlos Griñán Pérez. C/ Cobalto, 152. Hospitalet de Llobregat. Barcelona.

Vendo Software para Spectrum, tengo los últimos títulos (Space Harrier, Super Soccer... etc.). José Félix Garzón. C/ Tutulo, 2, 4.º dcha. 48007 Bilbao. Tl.: 94-446 55 16.

Cambio, compro y vendo todo tipo de programas para el Spectrum 48K Antonio Toribio Carreras. Polígono Pta. Madrid. Sector Málaga. Edif. Granada. Puerta B, 3.º izqda. 23740 Andújar. Jaén.

Vendo lote de más de 50 programas (juegos, utilidades, aplicaciones) por sólo 5.000 ptas. Para Spectrum. Miguel López. Sta. Cristina de Cobres. Vilaboia. Pontevedra. Tl.: 986-70 88 58. A partir de las 16 h.

Vendo programas para Spectrum, juegos y utilidades. Últimas novedades (Cobra, Uridium, Bounder, Asterix, etc.). Vendo Copión Turbo con instrucciones. Rafael Alcaide Jiménez. C/ Constitución, 20. Sant Feliu de Llobregat. Barcelona. Tl.: 93-666 00 21.

Vendo Spectrum 48K completo por 15.000 ptas. y regalo 77 programas originales y amplificador de sonido. Tl.: 93-354 62 21. Mañanas. Compro ordenador Sinclair QL. Ernesto López López. Calle Rebeco, 27, 2.º-1.º Barcelona 08031.

Vendo Spectrum 48K nuevo, con todos los accesorios (manual, cables). Incluyo interface y joystick, 30 juegos comerciales y «Tu primer libro del ZX Spectrum». Todo por 20.000. Miguel Morillo. Barcelona. Tl.: 93-333 14 88.

Vendo ordenador Dragon 64, monitor de fósforo verde, un joystick, 40 juegos,

un cassette, etc. Precio a convenir. Llamar a Miguel López. Tl.: 398 10 39. Barcelona (de 2 a 4 de la tarde).

Cambio juegos originales para Spectrum, poseo primeros títulos (Rambo, Bruce Lee, Saboteur, etc.). A mí me interesan Kun-Fu Master y Yie ar Kun-Fu 1 y 2. Miguel Herrera. Calle Formentera B.º D, 3.º-4.º Barcelona. Teléfono: 93-354 41 24.

Cambio programas para el Spectrum, desearía conseguir GHOSTS'N GOBLINS y BASKETBALL. Jorge Martínez. C/ Almaza, 27, 2.º C. Madrid 28011. Tl.: 463 47 37.

Vendo impresora ADMATE DP-80LQ nueva. Antonio Sáez Bravo. C/ Alberche, 136. 45007 Toledo. Tl.: 925-231 56 62.

Desearía intercambiar programas para Spectrum 48K. José Miguel Llop Juan. C/ Primavera, 8, 3.º Hospitalet de Llobregat. Barcelona.

Vendo o cambio juegos. Tengo últimas novedades (Paper Boy, 1942, Thanatos, Cobra, etc.). También vendo joystick Canon. Manuel María. Calle Trafalgar, 4, 2.º D. Algeciras. Cádiz. Teléfono: 956-66 41 81.

Intercambio juegos y utilidades para Spectrum. Dispongo de últimas novedades. En utilidades me interesan los lenguajes (Forth, Cobol, Logo...) y diseñadores gráficos. También compro Spectrum 48K con cables y manual por 10.000 ptas. Escribir a Juan Muñoz Falcó. Avenida Suecia, 4, 3.º 46010 Valencia.

Cambio contestador automático de llamadas Amper CM-52 125/220 con conector telefónico, en buen estado y los libros (Fotocopiados y encuadernados): Base II para el Amstrad PCW-8256 y CPC-6128; hoja electrónica de cálculo para el CPC 464 (Micro Spread) y una guía para el observador de estrellas (Star Watcher) para el CPC 464 por una impresora GP 50 S con Interface RS-232. También poseo programas para el 48K. Estoy dispuesto a mandar lista a quien la solicite. Ramón Varela Patón. Calle La Paz, 3, 1.º A. Cartagena. Murcia.

Vendo ordenador Einstein 80K de RAM (ampliables). Se incluye: monitor fósforo verde, impresora y programas de contabilidad, base de datos, tratamiento de textos, Multiplan. Precio a convenir. Llamar a los teléfonos 248 41 47-248 40 54 de Madrid y preguntar por Oscar.

LAS ESTRELLAS DE ZIGURAT:



Los autores de "SIR FRED" te presentan su nueva creación: Si buscas ACCIÓN sin límites y recorrer los paisajes africanos en una trepidante fuga, EL MISTERIO DEL NILO es tu vídeo-aventura.



NUCLEAR BOWLS



...ALERTA ROJA...ALERTA ROJA...
...ALERTA ROJA...ALERTA ROJA...
Queda poco tiempo para reparar el reactor de la central nuclear más potente de la Tierra. Aprovéchalo...



ZIGURAT SOFTWARE ESPAÑOL
Avda. Cardinal Herrera Oria, 163
28034 MADRID Tfno. (91) 201 84 69

Disponible SPECTRUM próximamente AMSTRAD y MSX

La máquina alucinante



EL ÚNICO
ORDENADOR
CON MILES Y MILES
DE PROGRAMAS
DISPONIBLES.

33.900 Ptas. + IVA



Al comprar
tu nuevo Spectrum
pide el Pasaporte Fantástico.
Podrás conseguir
un reloj alucinante.

Microprocesador Z80A. 128 K RAM. 32 K ROM. Teclado de 58 teclas.
32 columnas X 24 filas de texto. Gráficos de alta resolución
(256 X 192 pixels). 8 colores con dos niveles de brillo cada uno.
Calculadora en pantalla. 3 canales de sonido programables e
independientes. Cassette incorporada. Salida TV y monitor RGB.

Interface MIDI (Musical Instrument Digital Interface). Salida Serie RS 232
bidireccional. Dos conectores para joysticks. Conector plano
compatible con todos los modelos Spectrum anteriores. Editor de pantalla
y dos versiones BASIC en ROM. 48 K BASIC, compatible con Spectrum 16 K.
48 K y ZX + 128 K BASIC, compatible con ZX Spectrum 128

Nuevo **SINCLAIR ZX Spectrum +2**

C/ Aravaca, 22, 28040 Madrid. Tel. 459 30 01. Telex 47660 INSC E. Fax 459 22 92. Delegación en Cataluña: C/ Tarragona, 110. Tel. 325 10 58. 08015 Barcelona.